

TUGAS AKHIR

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
2024

UNDERGRADUATE THESIS

**INDONESIA RUGBY APP REBUILD WITH IONIC 7 AND
CAPACITOR**



Nathanael Adi Trianto

NPM: 6181901041

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2024**

ABSTRAK

Persatuan Rugby Union Indonesia merupakan Asosiasi Uni Rugbi Indonesia yang bertanggung jawab atas pengelolaan dan pengembangan Rugby Union di Indonesia. Rugby Indonesia memiliki website yang dibangun oleh PT Dnartworks Komunikasi Visual untuk memberikan informasi seputar kegiatan olahraga rugby di Indonesia. Pada sekitar tahun 2015, PT Dnartworks Komunikasi Visual membuat aplikasi Rugby Indonesia yang memanfaatkan Apache Cordova. Namun, saat ini aplikasi tersebut tidak dapat mengakses berita dari perangkat Android karena website sudah berubah dan Apache Cordova tidak lagi digunakan. Pengembangan aplikasi berbasis framework Ionic kini menggunakan Native Capacitor, yang memungkinkan Ionic untuk melepas ketergantungannya pada Angular dan Cordova.

Aplikasi Rugby Indonesia yang sedang dibangun saat ini dibuat dengan menggunakan Ionic 7 dan diimplementasikan dengan menggunakan library JavaScript dari React. Pada framework Ionic, terdapat UI Component yang memungkinkan developer untuk membangun user interface yang menarik dan responsif dengan cepat. Salah satu contoh dari UI Component tersebut adalah penggunaan tag <IonIcon> yang dapat menambahkan ikon ke dalam aplikasi. Selain itu, pada framework Ionic juga terdapat Capacitor Native yang berfungsi untuk mengakses fungsionalitas dari perangkat asli ke dalam aplikasi, sehingga pengguna dapat dengan mudah menggunakan aplikasi yang dibuat. Salah satu contoh dari Capacitor Native ini adalah penggunaan Capacitor Camera yang dapat melakukan penangkapan gambar. Dengan demikian, aplikasi yang dibuat dengan menggunakan Ionic 7 dan Capacitor pada environment web seperti HTML, CSS, dan JavaScript dapat juga dijalankan pada perangkat mobile seperti iOS dan Android. Namun, pengujian yang dilakukan pada tugas akhir ini hanya diuji pada perangkat Android saja.

Pengujian aplikasi dilakukan dalam dua tahap, yaitu pengujian fungsionalitas dan pengujian oleh pengguna umum yang dilakukan oleh enam responden. Responden diminta mengunduh aplikasi dari tautan Google Drive yang telah dibagikan dan mengisi kuesioner yang diberikan. Hasil yang didapatkan dari pengujian tersebut menunjukkan bahwa aplikasi berjalan dengan lancar pada perangkat Android.

Kata-kata kunci: Rugby Indonesia, Ionic 7, Capacitor, Android

ABSTRACT

Persatuan Rugby Union Indonesia is the Indonesian Rugby Union Association responsible for the management and development of Rugby Union in Indonesia. Rugby Indonesia has a website built by PT Dnartworks Komunikasi Visual to provide information about rugby activities in Indonesia. In around 2015, PT Dnartworks Komunikasi Visual created a Rugby Indonesia application that utilizes Apache Cordova. However, currently the application cannot access news from Android devices because the website has changed and Apache Cordova is no longer used. Application development based on the Ionic framework now uses Native Capacitor, which allows Ionic to release its dependency on Angular and Cordova.

The Indonesian Rugby application that is currently being built was created using Ionic 7 and implemented using the JavaScript library from React. In the Ionic framework, there is a UI Component that allows developers to build attractive and responsive user interfaces quickly. One example of the UI Component is the use of the <IonIcon> tag that can add icons to the application. In addition, the Ionic framework also has a Native Capacitor that functions to access functionality from native devices into the application, so that users can easily use the application created. One example of Capacitor Native is the use of Capacitor Camera that can capture images. Thus, applications made using Ionic 7 and Capacitor in web environments such as HTML, CSS, and JavaScript can also be run on mobile devices such as iOS and Android. However, the tests carried out in this final project were only tested on Android devices.

Application testing was carried out in two stages, namely functionality testing and testing by general users conducted by six respondents. Respondents were asked to download the application from the Google Drive link that had been distributed and fill out the questionnaire given. The results obtained from the test show that the application runs smoothly on Android devices.

Keywords: Indonesia Rugby, Ionic 7, Capacitor, Android

DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Metodologi	4
2 LANDASAN TEORI	5
2.1 Pembuatan Ulang	5
2.2 Aplikasi Rugby Indonesia	5
2.3 ReactJS [1]	6
2.3.1 Hooks	6
2.3.2 Components	7
2.3.3 Dangerously setting the inner HTML	9
2.4 Ionic 7 Framework [2]	9
2.4.1 UI Components	10
2.4.2 Capacitor Native	17
2.5 Arsitektur Ionic Framework	19
2.6 RSS [3]	20
2.7 React-Axios [4]	21
2.8 React Native XML to JSON Conversion [5]	22
2.9 Android Studio	22
3 ANALISIS	25
3.1 Analisis Sistem Kini	25
3.1.1 Halaman Utama	26
3.1.2 Halaman Teammate Photos	26
3.1.3 Halaman Rugby Clubs	26
3.1.4 Halaman Fixture & Results	27
3.2 Analisis Masalah Aplikasi Rugby Indonesia	27
3.3 Analisis RSS Rugby Indonesia	28
3.4 Analisis Sistem Usulan	28
3.4.1 Halaman Utama	29
3.4.2 Halaman Teammate Photos	29
4 PERANCANGAN	31
4.1 Perancangan Kelas	31
4.2 Perancangan Antarmuka	33
4.2.1 Perancangan Antarmuka Halaman <i>Latest News</i>	33

4.2.2	Perancangan Antarmuka Deskripsi Berita	34
4.2.3	Perancangan Antarmuka Halaman Teammate Photos	34
4.2.4	Perancangan Antarmuka Halaman <i>Frame</i>	35
4.2.5	Perancangan Antarmuka Menu	35
4.3	Perancangan Struktur HTML	35
4.3.1	<i>Header</i>	35
4.3.2	<i>Content</i>	36
4.4	Struktur Folder dan File Aplikasi	37
5	IMPLEMENTASI DAN PENGUJIAN	39
5.1	Spesifikasi Perangkat Keras	39
5.2	Implementasi Antarmuka	39
5.2.1	Implementasi Antarmuka <i>Latest News</i>	39
5.2.2	Implementasi Antarmuka <i>News Detail</i>	39
5.2.3	Implementasi Antarmuka Menu	40
5.2.4	Implementasi Antarmuka Halaman <i>Teammate Photos</i>	41
5.2.5	Implementasi Antarmuka Halaman <i>Frame</i>	41
5.3	Pengujian Fungsional	41
5.3.1	Pengujian Melihat Berita	42
5.3.2	Pengujian Melihat dan Melakukan <i>Upload</i> Gambar	42
5.4	Pengujian oleh Pengguna Umum	42
6	KESIMPULAN DAN SARAN	45
6.1	Kesimpulan	45
6.2	Saran	45
DAFTAR REFERENSI		47
A	KODE PROGRAM	49
B	HASIL KUESIONER ACCEPTANCE TEST	55

DAFTAR GAMBAR

1.1 Halaman aplikasi Rugby Indonesia	3
2.1 Gambar Hasil Action Sheet	11
2.2 Stack Teknologi Ionic Framework	19
2.3 Tampilan Antarmuka Android Studio	23
3.1 Use Case Diagram Aplikasi Rugby Indonesia Kini	25
3.2 Halaman Utama Aplikasi Rugby Indonesia	27
3.3 <i>Use Case Diagram Sistem Usulan Rugby Indonesia</i>	29
3.4 Analisis dari Halaman Utama Rugby Indonesia	29
3.5 Analisis dari Halaman Teammate Photos	30
4.1 Diagram Kelas Rugby Indonesia	31
4.2 Perancangan Antarmuka Bagian <i>Latest News</i>	33
4.3 Perancangan Antarmuka Bagian <i>Teammate Photos</i>	34
4.4 Perancangan Antarmuka Menu pada Aplikasi Rugby Indonesia	35
4.5 Struktur File Aplikasi Rugby Indonesia	37
5.1 Antarmuka Bagian <i>Latest News</i>	40
5.2 Halaman Menu pada Aplikasi Rugby Indonesia	40
5.3 Antarmuka Bagian <i>Teammate Photos</i>	41
B.1 Tangkapan Layar Pertama Hasil Kuesioner UAT	55
B.2 Tangkapan Layar Kedua Hasil Kuesioner UAT	56
B.3 Tangkapan Layar Ketiga Hasil Kuesioner UAT	57
B.4 Tangkapan Layar Keempat Hasil Kuesioner UAT	58

¹

BAB 1

²

PENDAHULUAN

³ 1.1 Latar Belakang

⁴ Dikutip dari buku Olahraga Rugby (2010) [6], *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 diperkenalkan di
⁸ Indonesia pada tahun 2004 dengan bantuan negara lain seperti Australia dan Selandia Baru sehingga
⁹ terbentuklah Indonesian Rugby Football Union (IRFU) yang bertujuan untuk mengembangkan
¹⁰ olahraga ini.

¹¹ Mengutip 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.

²⁰ PRUI juga memiliki website yang didirikan oleh PT Dnartworks Komunikasi Visual untuk mem-
²¹ berikan informasi seputar kegiatan olahraga rugby di Indonesia. Website ini dibangun menggunakan
²² platform WordPress⁴, sebuah Content Management System (CMS) yang memungkinkan pengguna
²³ untuk membuat dan mengelola website dengan mudah tanpa perlu pengetahuan dasar pemrograman.
²⁴ Dari website yang dimiliki oleh PRUI, dibangun juga sebuah aplikasi Rugby Indonesia menggunakan
²⁵ framework Ionic. Tujuan dari pembuatan aplikasi ini adalah untuk memudahkan pemain rugby
²⁶ dalam mengakses berita terbaru mengenai dunia rugby di Indonesia serta berbagi momen seru dari
²⁷ permainan rugby dengan mengunggah foto-foto dari aktivitas rugby yang sedang dilakukan.

²⁸ Dikutip dari GitHub Ionic [7], Ionic Framework merupakan *toolkit UI open-source* untuk
²⁹ membangun aplikasi modern, *cross-platform* yang berkualitas tinggi dari satu kode sumber dengan
³⁰ JavaScript dan web. Ionic menyediakan alat dan layanan untuk mengembangkan aplikasi *hybrid*
³¹ *mobile*, *desktop*, dan *progressive web* berdasarkan teknologi dan praktik pengembangan web modern,

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

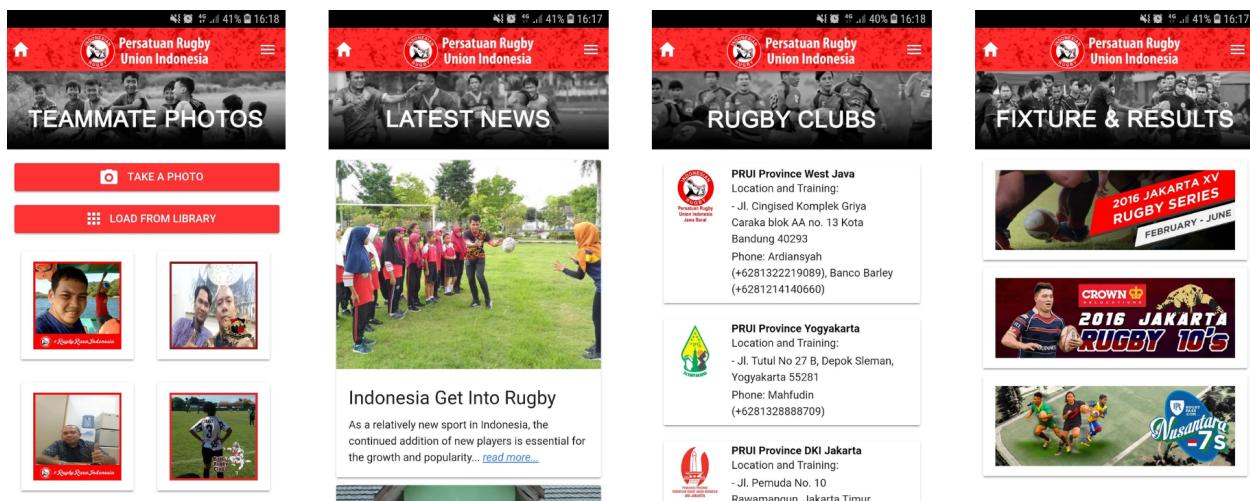
15 Capacitor [8], yang dikembangkan oleh tim Ionic, adalah runtime native open-source yang
16 menggantikan Apache Cordova. Capacitor memungkinkan pengembang untuk membangun aplikasi
17 mobile pada perangkat iOS dan Android serta aplikasi web progresif dengan menggunakan JavaScript,
18 HTML, dan CSS. Capacitor juga menyediakan cross-platform API yang dapat digunakan pada
19 berbagai platform dengan satu kode sumber.

20 RSS⁵ atau Really Simple Syndication adalah format sindikasi konten web yang merupakan
21 dialek dari XML. Semua file RSS harus disesuaikan dengan spesifikasi XML 1.0 yang dipublikasikan
22 oleh World Wide Web Consortium (W3C). Dokumen RSS terdiri dari elemen `<rss>` yang memiliki
23 atribut wajib bernama `version` untuk menentukan versi RSS yang digunakan. Di bawah elemen
24 `<rss>` terdapat elemen `<channel>` yang berisi informasi tentang saluran (metadata) dan isinya.

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

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

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



Gambar 1.1: Halaman aplikasi Rugby Indonesia dari Google Play Store

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

10 Aplikasi ini akan dibuat dengan memanfaatkan bantuan *framework* Ionic 7 dan Native Capacitor
 11 dengan memiliki:

- 12 • Halaman *Latest News*, di mana pengguna dapat melihat berita terbaru dari Rugby Indonesia.
- 13 • Halaman *Teammate Photos* di mana pengguna dapat mengambil foto dan langsung mengung-
 14 gahnya ke dalam halaman *teammate Photos*.

16 1.2 Rumusan Masalah

17 Rumusan masalah yang akan dibahas pada tugas akhir ini adalah bagaimana membangun ulang
 18 serta mengembangkan aplikasi Rugby Indonesia dengan memanfaatkan *framework* Ionic 7 dan
 19 Native Capacitor.

20 1.3 Tujuan

21 Tujuan yang ingin dicapai pada penulisan tugas akhir ini yaitu membangun ulang aplikasi Rugby
 22 Indonesia yang sudah memanfaatkan *framework* dari Ionic terbaru yaitu Ionic 7 dan juga Native
 23 Capacitor.

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

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

1.4 Batasan Masalah

- 2 Batasan masalah yang terdapat pada penggerjaan tugas akhir ini yaitu:
- 3 1. Aplikasi ini dibuat untuk perangkat Android saja, tidak untuk iOS. Sehingga pengujian dari
4 aplikasi ini hanya dilakukan pada platform berbasis Android saja.
- 5 2. Tidak terdapat *role* lain selain pengguna pada aplikasi ini.

1.5 Metodologi

- 7 Langkah-langkah yang dilakukan dalam penggerjaan tugas akhir ini yaitu:
- 8 1. Melakukan studi literatur serta mendalami ReactJS sebagai salah satu perpustakaan JavaScript
9 untuk membangun tampilan antar muka.
- 10 2. Melakukan studi literatur mengenai *framework* Ionic 7 dan juga Capacitor yang terdapat
11 pada Ionic Native.
- 12 3. Menganalisis terhadap aplikasi yang ada dan melakukan perancangan terhadap aplikasi yang
13 akan dibuat.
- 14 4. Membangun aplikasi Rugby Indonesia yang sudah menggunakan *framework* Ionic 7 serta
15 Capacitor.
- 16 5. Melakukan pengujian dan eksperimen.
- 17 6. Menulis dokumen tugas akhir.

1

BAB 2

2

LANDASAN TEORI

3 2.1 Pembuatan Ulang

4 Pembuatan ulang atau remanufacturing adalah proses membangun ulang produk sesuai spesifikasi
5 asli dengan menggunakan kombinasi komponen yang digunakan kembali, diperbaiki, dan baru.
6 Proses ini melibatkan perbaikan atau penggantian komponen yang sudah usang.[9]

7 Terdapat tiga buah jenis pembuatan ulang:

- 8 1. Remanufacturing tanpa kehilangan identitas: Produk yang ada dibangun di atas basis produk
9 sebelumnya dengan perbaikan dan peningkatan.
- 10 2. Remanufacturing dengan kehilangan identitas produk asli: Barang bekas dibongkar menjadi
11 komponen yang ditentukan sebelumnya dan dirakit kembali menjadi produk baru.
- 12 3. Remanufacturing berulang tanpa kehilangan identitas: Hasil akhir memiliki tampilan seperti
13 baru dan dilindungi oleh garansi yang sebanding dengan produk baru.

14 2.2 Aplikasi Rugby Indonesia

15 Aplikasi Rugby Indonesia merupakan aplikasi resmi dari Persatuan Rugby Union Indonesia yang
16 dapat memberikan informasi terbaru mengenai olahraga *rugby* di Indonesia. Aplikasi ini dapat
17 memberikan notifikasi langsung mengenai berita terakhir, turnamen yang akan datang, dan informasi
18 lainnya. Selain itu, aplikasi ini juga memungkinkan pengguna untuk mengambil gambar dan
19 menunjukkan dukungan mereka dengan rekan-rekan penggemar *rugby* lainnya dengan meng-upload
20 gambar ke dalam halaman *Teammate Photos*. Terdapat juga aplikasi multimedia pengenalan olahraga
21 *rugby* berbasis Android yang dapat memberikan informasi sejarah, peraturan, dan peralatan serta
22 tempat-latihan *rugby* di beberapa Kabupaten/Kota. Aplikasi Rugby Indonesia tersedia di Google
23 Play Store dan dapat diunduh secara gratis. Gambar dari halaman Rugby Indonesia dapat dilihat
24 pada Gambar 1.1.

25 Fitur-fitur yang ada pada aplikasi Rugby Indonesia dengan versi terakhir yang ada saat ini
26 yaitu:

- 27 1. Halaman *Latest News* yang diambil dari <https://rugbyindonesia.or.id/berita/> dengan
28 memanfaatkan protokol RSS.
- 29 2. Halaman *Fixture & Results*.
- 30 3. Halaman *Teammate Photos* dengan fungsi:
 - 31 • Pengguna dapat langsung mengambil foto dari aplikasi tersebut.
 - 32 • Pengguna dapat langsung memberikan *frame* terhadap foto tersebut.

- 1 • Pengguna dapat langsung mengunggah foto tersebut ke dalam galeri publik.
- 2 4. Halaman *Rugby Clubs* yang memiliki fungsi di mana pengguna dapat langsung mendaftar ke dalam *Rugby Clubs* yang berada di Indonesia.
- 3
- 4 5. Fungsi *Push Notifications*.

5 2.3 ReactJS [1]

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

11 2.3.1 Hooks

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

15 State Hooks

16 State Hooks adalah salah satu jenis Hooks yang memungkinkan pengembang untuk menggunakan
 17 *state* pada *function component* tanpa harus membuat sebuah kelas. Cara untuk menggunakan
 18 State Hooks terdapat pada Kode 2.1, di mana ‘useState’ digunakan untuk mendeklarasikan sebuah
 19 variabel *state* yang dapat langsung di *update*.

Kode 2.1: Contoh Potongan Kode State Hooks

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

25 Context Hooks

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

Kode 2.2: Contoh Potongan Kode Context Hooks

```
30 1 function Button() {
31 2   const theme = useContext(ThemeContext);
32 3   // ...
```

35 Ref Hooks

36 Ref Hook pada React adalah salah satu dari beberapa Hooks bawaan yang memungkinkan peng-
 37 gunaan *refs* dalam komponen fungsional. *Ref* digunakan untuk mengakses DOM atau nilai dari

- 1 elemen *child* dalam React. Cara untuk menggunakan Ref Hooks terdapat pada Kode 2.3. Pada
 2 kode tersebut, ‘useRef’ digunakan untuk mereferensikan sebuah nilai yang tidak digunakan dalam
 3 *rendering*.

Kode 2.3: Contoh Potongan Kode Ref Hooks

```
4
5 1 function Form() {
6 2   const inputRef = useRef(null);
7 3   // ...
```

9 Effect Hooks

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

Kode 2.4: Contoh Potongan Kode Effect Hooks

```
14
15 1 function ChatRoom({ roomId }) {
16 2   useEffect(() => {
17 3     const connection = createConnection(roomId);
18 4     connection.connect();
19 5     return () => connection.disconnect();
20 6   }, [roomId]);
21 7   // ...
```

23 Performance Hooks

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

Kode 2.5: Contoh Potongan Kode Performance Hooks

```
28
29 1 function TodoList({ todos, tab, theme }) {
30 2   const visibleTodos = useMemo(() => filterTodos(todos, tab), [todos, tab]);
31 3   // ...
32 4 }
```

34 2.3.2 Components

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

- 40 • <Fragment>
- 41 • <Profiler>
- 42 • <Suspense>
- 43 • <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*’. Fragment juga dapat digunakan untuk me-
 6 *render* beberapa elemen secara bersamaan tanpa harus menggunakan ‘<i>div</i>’. Ada dua cara untuk
 7 mendefinisikan Fragment, yaitu dengan menggunakan <React.Fragment> atau dengan menggunakan
 8 sintaksis singkat <>...</>. Cara untuk penggunaan komponen Fragment terdapat pada contoh
 9 Kode 2.6.

Kode 2.6: Contoh Potongan Kode Fragment

```
10
11 1 function Post() {
12 2   return (
13 3     <>
14 4       <PostTitle />
15 5       <PostBody />
16 6     </>
17 7   );
18 8 }
```

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

22 Profiler

23 Profiler adalah komponen yang digunakan untuk mengukur seberapa sering sebuah aplikasi React
 24 melakukan *rendering* dan seberapa besar biaya yang dikeluarkan untuk melakukan *rendering*
 25 tersebut. Cara penggunaan komponen Profiler terdapat pada contoh Kode 2.7.

Kode 2.7: Contoh Potongan Kode Profiler

```
26
27 1 <Profiler id="App" onRender={onRender}>
28 2   <App />
29 3 </Profiler>
```

31 Suspense

32 Suspense adalah fitur yang memungkinkan penundaan *render* komponen sampai data yang diper-
 33 lukan tersedia. Fitur ini berguna untuk meningkatkan responsivitas aplikasi dengan membiarkan
 34 komponen me-*render* terlebih dahulu sebelum data siap. Cara penggunaan komponen Suspense
 35 terdapat pada contoh Kode 2.8.

Kode 2.8: Contoh Potongan Kode Suspense

```
36
37 1 <Suspense fallback={<Loading />}>
38 2   <SomeComponent />
39 3 </Suspense>
```

41 StrictMode

42 StrictMode adalah sebuah komponen yang digunakan untuk menyoroti potensi masalah dalam
 43 sebuah aplikasi. Mode ini tidak berdampak dalam pembangunan produksi dan dapat diaktifkan

untuk bagian-bagian tertentu dalam aplikasi. StrictMode membantu dalam mengidentifikasi komponen-komponen dengan siklus hidup yang tidak aman dan melakukan berbagai pemeriksaan tambahan untuk turunannya. Mode ini berguna saat mengembangkan kode baru atau melakukan *debugging*. StrictMode dapat diterapkan pada bagian mana pun dalam aplikasi, bukan hanya pada keseluruhan aplikasi. Mode ini membantu dalam menulis kode React dengan cara yang lebih baik dengan memberikan peringatan terkait praktik terbaik. Mode ini juga dapat digunakan baik pada komponen fungsional maupun kelas. Namun, StrictMode me-*render* setiap komponen dalam aplikasi dua kali, sehingga sebaiknya hanya digunakan saat pengembangan atau *debugging*. Cara untuk menggunakan StrictMode terdapat pada contoh Kode 2.9. Pada kode tersebut, apabila terjadi error pada ‘`<App />`’, maka React akan memunculkan pesan error sebelum ‘`<App />`’ di-*render*.

Kode 2.9: Contoh Potongan Kode StrictMode

```
11 import { StrictMode } from 'react';
12 import { createRoot } from 'react-dom/client';
13
14 const root = createRoot(document.getElementById('root'));
15
16 root.render(
17   <StrictMode>
18     <App />
19   </StrictMode>
20 );
21
```

22 2.3.3 Dangerously setting the inner HTML

23 dangerouslySetInnerHTML merupakan sebuah objek dengan bentuk {__html:'<p>some html</p>'}
24 yang berisi string HTML mentah di dalamnya. Hal ini akan menimpa properti innerHTML dari
25 node DOM dan menampilkan HTML yang diberikan di dalamnya. Contoh kode penggunaan
26 dangerouslySetInnerHTML terdapat pada Kode 2.10. Pada kode tersebut, pengembang harus
27 memastikan bahwa kode HTML tersebut aman atau berasal dari sumber yang terpercaya.

Kode 2.10: Contoh Potongan Kode dangerouslySetInnerHTML

```
28
29 1 const markup = { __html: '<p>some raw html</p>' };
30 2 return <div dangerouslySetInnerHTML={markup} />;
```

32 2.4 Ionic 7 Framework [2]

³³ Ionic 7 adalah sebuah *framework* untuk membangun aplikasi *mobile hybrid* menggunakan HTML5,
³⁴ CSS, dan JavaScript. Ionic 7 mendukung Angular 14+, React 17+, dan Vue 3.0.6+. *Framework* ini
³⁵ dapat digunakan secara gratis dan juga bersifat *open-source*, baik digunakan oleh pribadi maupun
³⁶ komersial. Untuk menggunakan Ionic 7, pengembang perlu meng-*install* dahulu Ionic tersebut
³⁷ dengan cara seperti yang terdapat pada Kode 2.11.

Kode 2.11: Kode untuk meng-install Ionic

38
39 1 npm install -g @ionic/cli

41 Setelah pengembang meng-*install* Ionic, pengembang dapat mengetikan *command ionic start*
42 untuk mulai membuat aplikasi menggunakan Ionic. Setelah aplikasi dibuat, pengembang dapat
43 menjalankan aplikasi tersebut dengan menggunakan *command ionic serve*.

2.4.1 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:

Action Sheet

Action Sheet merupakan sebuah komponen yang berguna untuk memunculkan dialog. Dialog tersebut akan melakukan pemberhentian sementara terhadap aplikasi yang sedang dijalankannya dan pengguna harus memilih pilihan yang berada di dalam dialog tersebut. Cara penggunaan dari Action Sheet terdapat pada Kode 2.12.

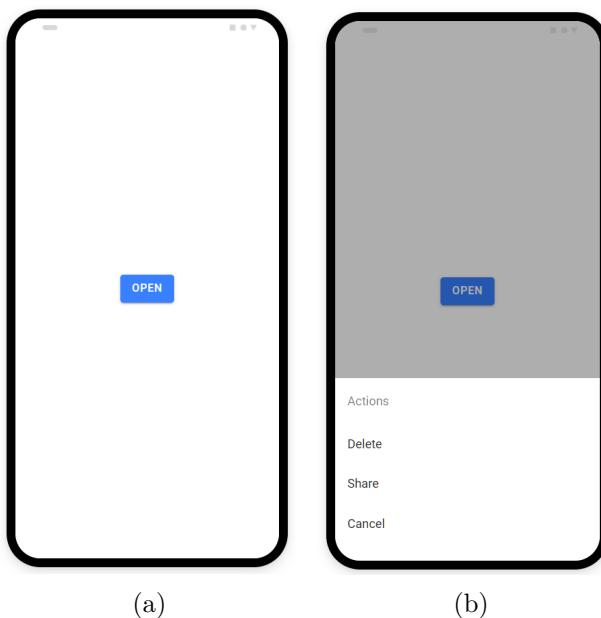
Kode 2.12: Contoh Potongan Kode Action Sheet

```

11 import React from 'react';
12 import { IonActionSheet, IonButton } from '@ionic/react';
13
14 function Example() {
15   return (
16     <>
17       <IonButton id="open-action-sheet">Open</IonButton>
18       <IonActionSheet
19         trigger="open-action-sheet"
20         header="Actions"
21         buttons={[
22           {
23             text: 'Delete',
24             role: 'destructive',
25             data: {
26               action: 'delete',
27             },
28           },
29           {
30             text: 'Share',
31             data: {
32               action: 'share',
33             },
34           },
35           {
36             text: 'Cancel',
37             role: 'cancel',
38             data: {
39               action: 'cancel',
40             },
41           },
42         ]}
43       ></IonActionSheet>
44     </>
45   );
46 }
47 export default Example;

```

Hasil yang ditampilkan dari Kode 2.12 terdapat pada Gambar 2.1.



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

¹ Properti ‘role’ pada Action Sheet adalah sebuah properti yang diberikan untuk memberikan *role*
² ‘cancel’ atau ‘destructive’ pada tombol yang berada di dalam Action Sheet. Nilai ‘cancel’ digunakan
³ untuk tombol yang akan membatalkan aksi yang dilakukan, sedangkan nilai ‘destructive’ digunakan
⁴ untuk tombol yang akan menghapus atau mengubah data yang ada. Selain properti role, Action
⁵ Sheet memiliki properti yang lain, properti tersebut terdapat 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 action-sheet	String atau undefined
3	backdrop-dissmiss	Menutup <i>action-sheet</i> apabila backdrop diklik	'true' atau 'false'

6 Button

7 Button merupakan elemen interaktif yang dapat digunakan dalam berbagai aplikasi untuk menyedi-
8 akan fitur tombol standar. Cara penggunaan komponen Button terdapat pada Kode 2.13.

Kode 2.13: Contoh Potongan Kode Button

```
9 <IonButton>Open</IonButton>  
10 1
```

12 Pada komponen Button terdapat properti ‘expand’ dan ‘icons’. Secara *default*, komponen
13 Button memiliki *style display: inline-block* dan tidak memiliki ikon. Namun dengan properti
14 *expand*, *style* pada komponen Button dapat diubah dengan cara memberikan properti ‘expand’ pada
15 komponen Button seperti <IonButton expand="block">Block</IonButton>. Properti ikon juga
16 dapat ditambahkan pada awal, akhir, ataupun hanya terdapat ikon pada Button tersebut, seperti

- 1 pada Kode 2.14. Saat menambahkan Icon, pengembang bisa mengikuti cara yang terdapat pada
 2 subbab 2.4.1.

Kode 2.14: Contoh Potongan Kode Button Menggunakan Icon

```
41 <IonButton>
42   <IonIcon slot="start" icon={star}></IonIcon>
43   Left Icon
44 </IonButton>
```

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

11 Card

12 Card merupakan komponen yang digunakan untuk menampilkan konten seperti teks, gambar,
 13 tombol, dan daftar dalam sebuah kotak. Komponen ini biasanya terdiri dari *header*, judul, gambar,
 14 dan konten utama. Card dapat digunakan sebagai komponen tunggal atau digabungkan dengan
 15 komponen lain untuk membuat tampilan yang lebih kompleks. Card dapat disesuaikan dengan
 16 menggunakan properti CSS seperti ‘background’ dan ‘color’. Cara menggunakan Card terdapat
 17 pada contoh Kode 2.15.

Kode 2.15: Contoh Potongan Kode Card

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

36 Pengembang juga bisa menggunakan Card sebagai Media Card dengan menambahkan elemen
 37 pada Card tersebut ataupun Button Card dengan menambahkan <IonButton> pada Card
 38 tersebut.

39 Content

40 Content merupakan komponen yang berguna untuk menyediakan area konten yang dapat dikontrol
 41 dan diubah menggunakan CSS. Dalam satu tampilan hanya terdapat satu konten. Konten dan
 42 komponen Ionic lainnya dapat dikostumisasi ulang dengan menggunakan CSS yang tersedia. Cara
 43 menggunakan Card terdapat pada contoh Kode 2.16.

Kode 2.16: Contoh Potongan Kode Content

```
44
45 import React from 'react';
46 import { IonContent } from '@ionic/react';
47
48 function Example() {
49   return (
50     <IonContent className="ion-padding">
51       <h1>Heading 1</h1>
52   
```

```

21   <p>Here's a small text description for the content. Nothing more, nothing less.</p>
22 </IonContent>
23 );
24 }
25 export default Example;

```

7 Pada Content, komponen ini dapat ditambahkan header, footer. Lalu komponen content ini juga
 8 dapat berupa Fixed Content dan juga Fullscreen Content. Secara *default*, Content akan memenuhi
 9 *header* dan *footer*.

10 Grid

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

16 Icon

17 Komponen Icon adalah elemen dasar yang tersedia melalui *library* Ionicons. Secara *default*, Icon
 18 dipersiapkan dengan semua aplikasi Ionic Framework. Komponen ini dapat digunakan untuk
 19 menampilkan ikon dari set Ionicons atau ikon yang berupa SVG. Selain itu, komponen ini juga
 20 mendukung pengaturan seperti ukuran dan warna. Cara penggunaan Icon terdapat pada Kode 2.17.

Kode 2.17: Contoh Kode Penggunaan Icon

```

21 import React from 'react';
22 import { IonIcon } from '@ionic/react';
23 import { logoIonic } from 'ionicons/icons';
24
25 function Example() {
26   return (
27     <>
28       <IonIcon icon={logoIonic}></IonIcon>
29       <IonIcon icon={logoIonic} size="large"></IonIcon>
30       <IonIcon icon={logoIonic} color="primary"></IonIcon>
31       <IonIcon icon={logoIonic} size="large" color="primary"></IonIcon>
32     </>
33   );
34 }
35 export default Example;

```

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

41 List

42 List digunakan untuk menampilkan data dalam bentuk baris, seperti daftar kontak, daftar putar,
 43 atau menu. Komponen ini mendukung berbagai macam interaksi, termasuk menggeser *item* untuk
 44 menampilkan opsi, menarik untuk menyusun ulang *item* dalam daftar, dan menghapus *item*.
 45 Komponen ini dapat ditambah berbagai elemen ke dalam daftar, seperti teks, tombol, ikon, dan
 46 gambar dengan ukuran yang kecil. Cara penggunaan List terdapat pada Kode 2.18.

Kode 2.18: Contoh Kode Penggunaan List

```

11 import React from 'react';
12 import { IonItem, IonLabel, IonList } from '@ionic/react';
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;

```

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

32 Menu

33 Menu merupakan *navigation drawer* yang meluncur dari sisi tampilan saat dijalankan. Komponen
 34 ini adalah pola navigasi umum yang dapat secara permanen ditampilkan di layar saat diperlukan.
 35 Komponen Menu dapat digunakan untuk membuat tata letak aplikasi yang lebih terstruktur dan
 36 meningkatkan pengalaman pengguna dengan menyediakan akses mudah ke berbagai bagian dari
 37 aplikasi. Cara penggunaan Menu terdapat pada Kode 2.19.

Kode 2.19: Contoh Kode Penggunaan Menu

```

38
39 1 import React from 'react';
40 2 import { IonButtons, IonContent, IonHeader, IonMenu, IonMenuButton, IonPage, IonTitle, IonToolbar } from '@ionic/react';
41 3 function Example() {
42 4   return (
43 5     <>
44 6       <IonMenu contentId="main-content">
45 7         <IonHeader>
46 8           <IonToolbar>
47 9             <IonTitle>Menu Content</IonTitle>
48 0           </IonToolbar>
49 1         </IonHeader>
50 2         <IonContent className="ion-padding">This is the menu content.</IonContent>
51 3       </IonMenu>
52 4       <IonPage id="main-content">
53 5         <IonHeader>
54 6           <IonToolbar>
55 7             <IonButtons slot="start">

```

```

18         <IonMenuButton></IonMenuButton>
19     </IonButtons>
20     <IonTitle>Menu</IonTitle>
21   </IonToolbar>
22 </IonHeader>
23 <IonContent className="ion-padding">Tap the button in the toolbar to open the menu.</IonContent>
24 </IonPage>
25   </>
26 );
27 }
28 export default Example;

```

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

18 Modal

19 Modal dapat digunakan langsung ke dalam template. Ini mengurangi jumlah handler yang perlu
 20 dihubungkan untuk menampilkan modal.

21 Saat menggunakan Modal dengan Angular, React, atau Vue, komponen yang diberikan akan
 22 dihancurkan ketika modal ditutup. Fungsi ini disediakan oleh kerangka kerja JavaScript. Jika
 23 menggunakan modal tanpa kerangka kerja JavaScript, komponen yang diberikan tidak akan
 24 dihancurkan. Jika fungsi ini diperlukan, disarankan menggunakan modalController.

25 Contoh penggunaan Modal terdapat pada Kode 2.20. Pada kode tersebut, Modal menggunakan
 26 properti `isOpen`. `isOpen` menggunakan binding data satu arah, yang berarti `isOpen` tidak akan
 27 otomatis disetel ke false saat modal ditutup. Pengembang harus mendengarkan event `ionModa-`
`IDidDismiss` atau `didDismiss` dan mengatur `isOpen` ke false. Ini untuk mencegah bagian internal
 28 `ion-modal` terlalu terkait dengan status aplikasi. Dengan binding data satu arah, modal hanya perlu
 29 memperhatikan nilai boolean yang diberikan oleh variabel reaktif. Dengan binding data dua arah,
 30 modal perlu memperhatikan nilai boolean serta keberadaan variabel reaktif itu sendiri, yang dapat
 31 menyebabkan perilaku non-deterministik dan membuat aplikasi lebih sulit untuk di-debug.

Kode 2.20: Contoh Kode Penggunaan Modal

```

33
34 1 import React, { useState } from 'react';
35 2 import { IonButtons, IonButton, IonModal, IonHeader, IonContent, IonToolbar, IonTitle, IonPage } from '@ionic/react';
36 3
37 4 function Example() {
38 5   const [isOpen, setIsOpen] = useState(false);
39 6
40 7   return (
41 8     <IonPage>
42 9       <IonHeader>
43 10         <IonToolbar>
44 11           <IonTitle>Inline Modal</IonTitle>
45 12         </IonToolbar>
46 13       </IonHeader>
47 14       <IonContent className="ion-padding">
48 15         <IonButton expand="block" onClick={() => setIsOpen(true)}>
49 16           Open
50 17         </IonButton>
51 18         <IonModal isOpen={isOpen}>
52 19           <IonHeader>
53 20             <IonToolbar>
54 21               <IonTitle>Modal</IonTitle>
55 22               <IonButtons slot="end">
56 23                 <IonButton onClick={() => setIsOpen(false)}>Close</IonButton>
57 24               </IonButtons>
58 25           </IonHeader>

```

```

26     </IonHeader>
27     <IonContent className="ion-padding">
28       <p>
29         Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni illum quidem recusandae ducimus quos
30         reprehenderit. Veniam, molestias quos, dolorum consequuntur nisi deserunt omnis id illo sit cum qui.
31         Eaque, dicta.
32       </p>
33     </IonContent>
34   </IonModal>
35   </IonContent>
36 </IonPage>
37 );
38 }
39 export default Example;

```

Metode yang terdapat pada Modal yaitu:

- dismiss
 - Deskripsi: Menutup modal setelah ditampilkan.
 - Signature: dismiss(data?: any, role?: string) => Promise<boolean>
- getCurrentBreakpoint
 - Deskripsi: Mengembalikan breakpoint saat ini dari modal gaya lembar.
 - Signature: getCurrentBreakpoint() => Promise<number | undefined>
- onDidDismiss
 - Deskripsi: Mengembalikan promise yang diselesaikan saat modal telah ditutup.
 - Signature: onDidDismiss<T = any>() => Promise<OverlayEventDetail<T>>
- onWillDismiss
 - Deskripsi: Mengembalikan promise yang diselesaikan saat modal akan ditutup.
 - Signature: onWillDismiss<T = any>() => Promise<OverlayEventDetail<T>>
- present
 - Deskripsi: Menampilkan modal setelah dibuat.
 - Signature: present() => Promise<void>
- setCurrentBreakpoint
 - Deskripsi: Memindahkan modal gaya lembar ke breakpoint tertentu. Nilai breakpoint harus sesuai dengan yang didefinisikan dalam array breakpoints.
 - Signature: setCurrentBreakpoint(breakpoint: number) => Promise<void>

37 Router

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

44 Toolbar

Toolbar merupakan komponen yang digunakan untuk menampilkan judul, tombol, ikon, tombol kembali, tombol menu, kotak pencarian, segmen, dan indikator proses di aplikasi. Toolbars umumnya ditempatkan di atas atau di bawah konten dan menyediakan konten dan tindakan untuk layar saat

- 1 ini. Ketika toolbar ditempatkan di *header*, Toolbar akan muncul di bagian atas konten, sedangkan
 2 jika ditempatkan di *footer*, Toolbar akan muncul di bagian bawah. Cara menggunakan Toolbar
 3 terdapat pada Kode 2.21.

Kode 2.21: Contoh Kode Penggunaan Toolbar

```

51 import React from 'react';
52 import { IonFooter, IonHeader, IonTitle, IonToolbar } from '@ionic/react';
53
54 function Example() {
55   return (
56     <>
57       <IonHeader>
58         <IonToolbar>
59           <IonTitle>Header Toolbar</IonTitle>
60         </IonToolbar>
61       </IonHeader>
62
63       <IonFooter>
64         <IonToolbar>
65           <IonTitle>Footer Toolbar</IonTitle>
66         </IonToolbar>
67       </IonFooter>
68     </>
69   );
70 }
71 export default Example;

```

- 27 Pada kode tersebut, Header Toolbar akan diletakan pada bagian *header* dan Footer Toolbar
 28 akan diletakan pada bagian *footer*.

29 2.4.2 Capacitor Native

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

35 Camera

- 36 Plugin Camera pada Ionic 7 adalah sebuah plugin yang memungkinkan pengguna untuk mengambil
 37 foto dengan kamera atau memilih foto yang sudah ada dari album foto. Plugin ini dapat di-
 38 *install* dengan perintah `npm install @capacitor/camera` dan `npx cap sync`, serta menambahkan
 39 beberapa izin pada file ‘Info.plist’ untuk iOS dan ‘AndroidManifest.xml’ untuk Android. Selain itu,
 40 *plugin* ini juga memerlukan PWA Elements agar dapat berfungsi. Cara untuk meng-*install* Camera
 41 terdapat pada Kode 2.22, sedangkan cara untuk menggunakan Camera terdapat pada contoh Kode
 42 2.23.

Kode 2.22: Kode untuk Menginstal Plugin Camera

```

43 1 npm install @capacitor/camera
44

```

Kode 2.23: Contoh Kode Capacitor Camera

```

46 import { Camera, CameraResultType } from '@capacitor/camera';
47
48 const takePicture = async () => {
49   const image = await Camera.getPhoto({
50     quality: 90,
51

```

```

16     allowEditing: true,
27   resultType: CameraResultType.Uri
38 });
49
50 // image.webPath will contain a path that can be set as an image src.
51 // You can access the original file using image.path, which can be
52 // passed to the Filesystem API to read the raw data of the image,
53 // if desired (or pass resultType: CameraResultType.Base64 to getPhoto)
54 var imageUrl = image.webPath;
55
56 // Can be set to the src of an image now
57 imageElement.src = imageUrl;
58 };

```

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

21 **Filesystem**

22 Filesystem API menyediakan alat seperti NodeJS untuk bekerja dengan *file* pada perangkat.
 23 Pengembang dapat menggunakan *plugin* ini untuk melakukan operasi *file* umum seperti membaca,
 24 tulis, dan mengelola isi direktori. Cara untuk meng-*install* Filesystem terdapat pada Kode 2.24,
 25 sedangkan cara untuk menggunakan Filesystem terdapat pada contoh Kode 2.25.

Kode 2.24: Kode untuk Menginstal Plugin Filesystem

```

26
27 1 npm install @capacitor/actionsheet

```

Kode 2.25: Contoh Kode Penggunaan Filesystem

```

29
30 1 import { Filesystem, Directory, Encoding } from '@capacitor/filesystem';
31 2
32 3 const writeSecretFile = async () => {
33 4   await Filesystem.writeFile({
34 5     path: 'secrets/text.txt',
35 6     data: 'This is a test',
36 7     directory: Directory.Documents,
37 8     encoding: Encoding.UTF8,
38 9   });
39 0 };
40 1
41 2 const readSecretFile = async () => {
42 3   const contents = await Filesystem.readFile({
43 4     path: 'secrets/text.txt',
44 5     directory: Directory.Documents,
45 6     encoding: Encoding.UTF8,
46 7   });
47 8
48 9   console.log('secrets:', contents);
49 0 };
50 1
51 2 const deleteSecretFile = async () => {
52 3   await Filesystem.deleteFile({
53 4     path: 'secrets/text.txt',
54 5     directory: Directory.Documents,
55 6   });
56 7 };
57 8
58 9 const readFilePath = async () => {
59 0   // Here's an example of reading a file with a full file path. Use this to
59 1   // read binary data (base64 encoded) from plugins that return File URIs, such as
59 2   // the Camera.
59 3   const contents = await Filesystem.readFile({
59 4     path: 'file:///var/mobile/Containers/Data/Application/22A433FD-D82D-4989-8BE6-9FC49DEA20BB/Documents/text.txt',
59 5   });
59 6 };

```

```

B6
27   console.log('data:', contents);
28 }

```

5 Preference

- 6 Plugin Preference adalah *plugin* yang berguna untuk menyimpan data sederhana dalam bentuk kunci
 7 atau nilai yang dapat diakses secara bersamaan. Preferences API menyediakan area penyimpanan
 8 data yang mendukung kunci atau nilai untuk aplikasi Ionic. Untuk menggunakan Plugin Preference
 9 dalam aplikasi Ionic, perlu meng-*install plugin* Capacitor Preferences dengan cara yang terdapat
 10 pada Kode 2.26, sedangkan cara untuk menggunakan Capacitor terdapat pada Kode 2.27.

Kode 2.26: Kode untuk Menginstal Plugin Preference

```

11 npm install @capacitor/preferences
12

```

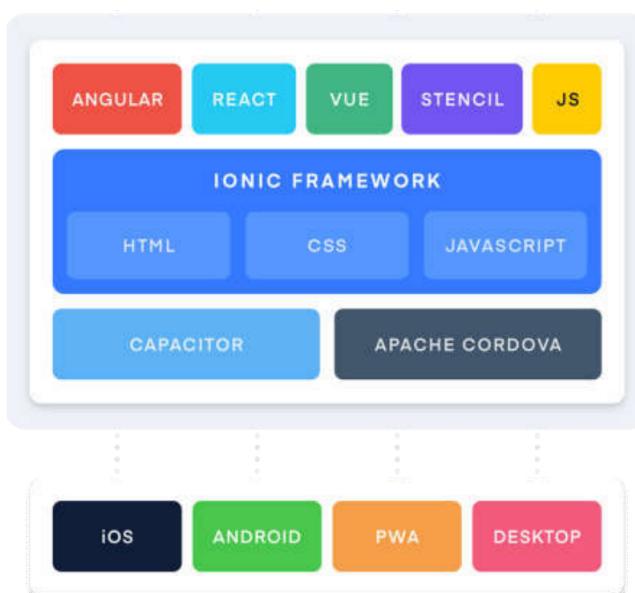
Kode 2.27: Contoh Kode Plugin Preference

```

14 import { Preferences } from '@capacitor/preferences';
15
16 const setName = async () => {
17   await Preferences.set({
18     key: 'name',
19     value: 'Max',
20   });
21
22 };
23
24 const checkName = async () => {
25   const { value } = await Preferences.get({ key: 'name' });
26
27   console.log(`Hello ${value}!`);
28 };
29
30 const removeName = async () => {
31   await Preferences.remove({ key: 'name' });
32 };
33
34

```

34 2.5 Arsitektur Ionic Framework



Gambar 2.2: Stack Teknologi Ionic Framework

1 Dari Gambar 2.2, arsitektur yang digunakan oleh Ionic Framework terdiri dari beberapa lapisan yang
 2 saling berhubungan untuk membentuk aplikasi mobile hybrid yang dapat dijalankan di berbagai
 3 platform seperti Android, iOS, dan Windows Phone. Berikut adalah penjelasan tentang arsitektur
 4 Ionic Framework:

5 1. Lapisan Utama:

- 6 • Android Runtime: Ionic menggunakan Dalvik Virtual Machine (DVM) yang dirancang
 7 khusus untuk Android.
- 8 • iOS Runtime: Ionic menggunakan mesin virtual yang sesuai dengan iOS.
- 9 • Windows Runtime: Ionic menggunakan mesin virtual yang sesuai dengan Windows.

10 2. Lapisan Framework:

- 11 • Ionic Framework: Ionic Framework adalah SDK open-source yang dibangun di atas AngularJS dan Apache Cordova. Ini menyediakan alat dan layanan untuk mengembangkan
 12 aplikasi mobile hybrid menggunakan teknologi web seperti HTML, CSS, dan JavaScript.
- 13 • Apache Cordova: Cordova menyediakan akses ke fitur perangkat native melalui API.

14 3. Lapisan Aplikasi:

- 15 • Aplikasi Mobile Hybrid: Aplikasi yang dibuat dengan Ionic Framework dapat dijalankan
 16 di berbagai platform mobile dan memiliki tampilan dan interaksi yang mirip dengan
 17 aplikasi native.

18 Arsitektur ini memungkinkan Ionic untuk mengembangkan aplikasi mobile yang dapat dijalankan
 19 di berbagai platform dengan menggunakan teknologi web yang familiar dan akses ke fitur perangkat
 20 native melalui API.

22 **2.6 RSS [3]**

23 RSS merupakan singkatan dari Really Simple Syndication adalah format sindikasi konten web yang
 24 merupakan dialek dari XML. Semua file RSS harus disesuaikan dengan spesifikasi XML 1.0 yang
 25 dipublikasikan oleh World Wide Web Consortium (W3C). Dokumen RSS terdiri dari elemen <rss>
 26 yang memiliki atribut wajib bernama version untuk menentukan versi RSS yang digunakan. Di
 27 bawah elemen <rss> terdapat elemen <channel> yang berisi informasi tentang saluran (metadata)
 28 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.

2.7 React-Axios [4]

Package React-Axios merupakan komponen React yang menggunakan Axios untuk melakukan permintaan asinkron di dalam render. Fitur dari React-Axios yaitu:

- Mewarisi semua fitur utama dari Axios.
- Berbasis komponen React.
- Menggunakan callback fungsi anak untuk menangani hasil permintaan (error, response, isLoading, makeRequest, axios).
- Membatalkan permintaan sebelumnya secara otomatis.
- Debounce untuk mencegah permintaan berulang dengan cepat.
- Permintaan hanya dipicu saat ada perubahan properti dan dalam keadaan siap (isReady).
- Mendukung callback untuk sukses (onSuccess), kesalahan (onError), dan loading (onLoading).
- Mendukung instance Axios kustom melalui properti atau <AxiosProvider>.

Cara untuk melakukan install package React-Axios terdapat pada Kode 2.28.

Kode 2.28: Kode untuk Menginstall Komponen React-Axios

```
14 // npm install react-axios
```

Komponen yang dimiliki oleh React-Axios meliputi komponen dasar dan juga komponen pembantu.

- Komponen dasar: <Request> merupakan komponen dasar untuk melakukan permintaan HTTP dengan berbagai properti (instance, method, url, data, params, config, debounce, debounceImmediate, isReady, onSuccess, onLoading, onError).
- Komponen bantu: Komponen bantu yang terdapat pada React-Axios yaitu <Get>, <Delete>, <Head>, <Post>, <Put>, <Patch>. Komponen ini berfungsi untuk permintaan HTTP yang spesifik.

Contoh penggunaan React-Axios terdapat pada Kode 2.29.

Kode 2.29: Contoh Kode dalam Melakukan GET Menggunakan React-Axios

```
26 // Post a request for a user with a given ID
27 1 render() {
28 2   return (
29 3     <div>
30 4       <Get url="/api/user" params={{id: "12345"}}>
31 5         {(error, response, isLoading, makeRequest, axios) => {
32 6           if(error) {
33 7             return (<div>Something bad happened: {error.message} <button onClick={() => makeRequest({ params: { reload: true } })}>Retry</button></div>)
34 8           }
35 9           else if(isLoading) {
36 10             return (<div>Loading...</div>)
37 11           }
38 12           else if(response !== null) {
39 13             return (<div>{response.data.message} <button onClick={() => makeRequest({ params: { refresh: true } })}>Refresh</button></div>)
40 14           }
41 15           return (<div>Default message before request is made.</div>)
42 16         }
43 17       </Get>
44 18     </div>
45 19   )
46 20 }
```

1 2.8 React Native XML to JSON Conversion [5]

- 2 Package React Native XML to JSON Conversion merupakan konverter sederhana dari XML ke
 3 objek JavaScript dan mendukung konversi dua arah menggunakan sax-js dan xmlbuilder-js. Package
 4 ini diciptakan dari hasil fork pada node-xml2js yang mendukung React Native sehingga mendukung
 5 platform Android dan juga iOS.
- 6 Cara untuk melakukan install package React-Native-XML2JS terdapat pada Kode 2.30.

Kode 2.30: Kode untuk Menginstal Komponen React-Axios

```
7 81 npm install react-native-xml2js
```

- 10 Terdapat juga contoh penggunaan XML2JS terdapat pada Kode 2.31.

Kode 2.31: Kode Contoh Penggunaan XML2JS

```
11 121 var parseString = require('react-native-xml2js').parseString;  

132 var xml = "<root>Hello_xml2js!</root>"  

143 parseString(xml, function (err, result) {  

154   console.dir(result);  

165 });
```

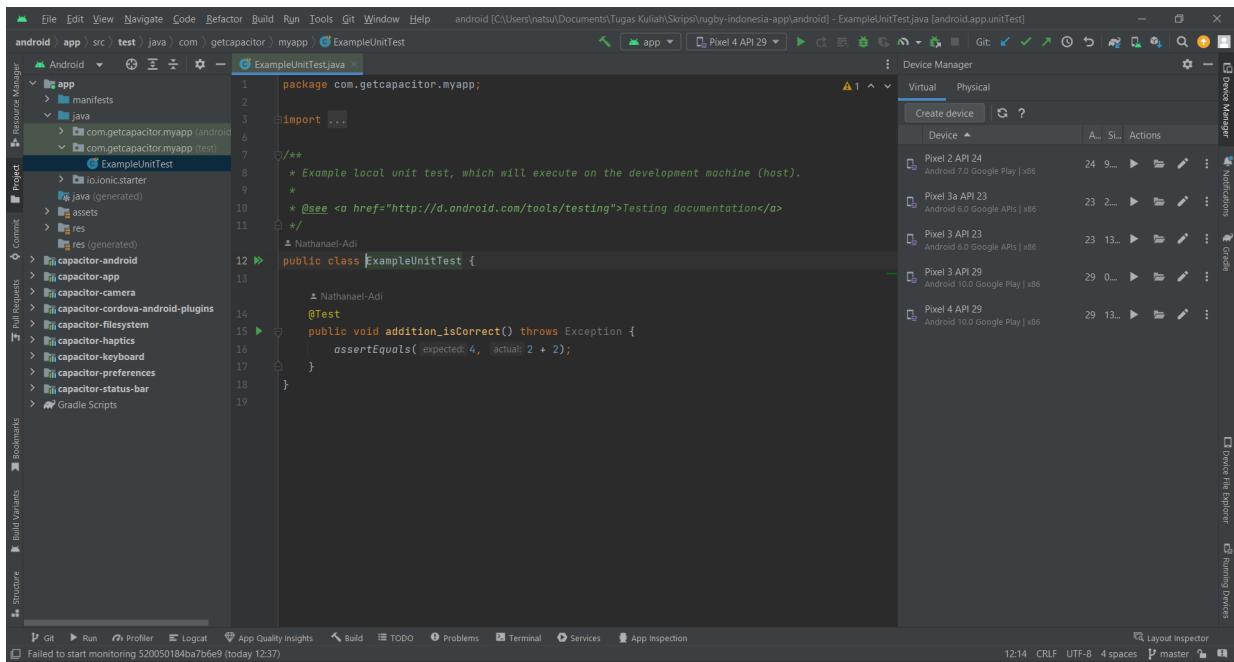
18 2.9 Android Studio

- 19 Android Studio¹ adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan
 20 aplikasi Android yang dikembangkan oleh Google dan berbasis pada IntelliJ IDEA. IDE ini mena-
 21 warkan berbagai fitur yang memudahkan dan mempercepat proses pengembangan aplikasi Android,
 22 fitur-fitur tersebut yaitu:

- 23 1. Sistem *build* berbasis Gradle yang fleksibel.
- 24 2. Emulator yang kaya fitur.
- 25 3. Lingkungan pengembangan yang terpadu untuk semua perangkat Android.
- 26 4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- 27 5. *Template* kode dan integrasi GitHub untuk membuat fitur aplikasi yang umum dan mengimpor
kode sampel.
- 28 6. Alat penguji dan kerangka kerja yang ekstensif.
- 29 7. Fitur lint untuk meningkatkan kinerja, kegunaan, dan kompatibilitas versi.
- 30 8. Dukungan C++ dan NDK.
- 31 9. Dukungan bawaan untuk Google Cloud Platform.

33 Android Studio digunakan untuk mendesain, membangun, menjalankan, dan menguji aplikasi
 34 untuk platform Android. IDE ini tersedia untuk platform desktop Mac, Windows, dan Linux.
 35 Tampilan dari Android Studio terdapat pada Gambar 2.3.

¹<https://developer.android.com/studio/intro>



Gambar 2.3: Tampilan Antarmuka Android Studio

1

BAB 3

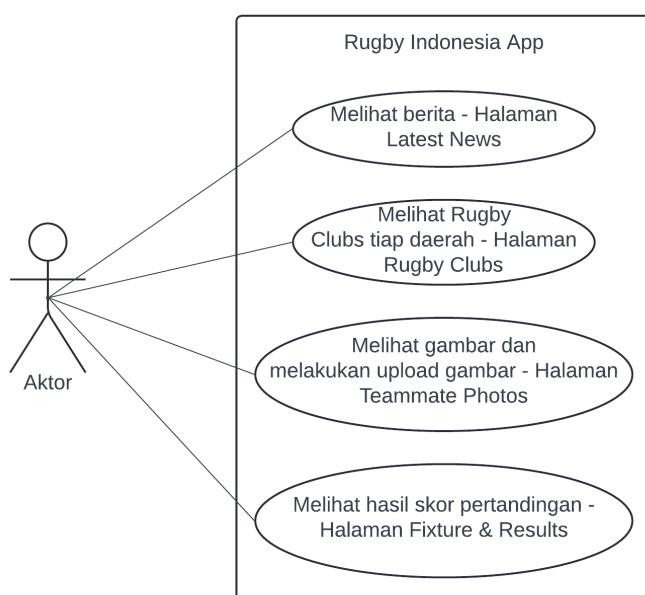
2

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 Photos*, 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 Aplikasi Rugby Indonesia Kini

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

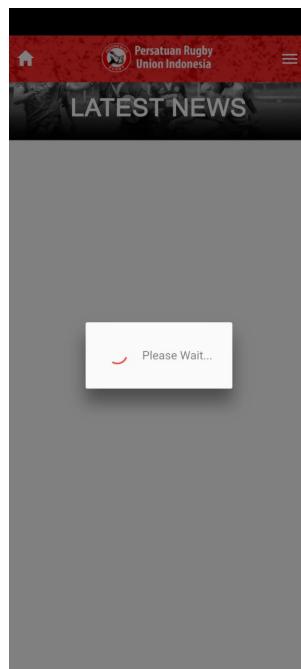
- ² Pada halaman Fixture & Results, pengguna dapat melihat hasil pertandingan skor dari permainan
³ rugby yang sudah diselenggarakan. Skenario pengguna saat berada pada halaman Fixture & Results
⁴ terdapat pada Tabel 3.4.

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 Kode 3.1.

Kode 3.1: Potongan Kode RSS Rugby Indonesia

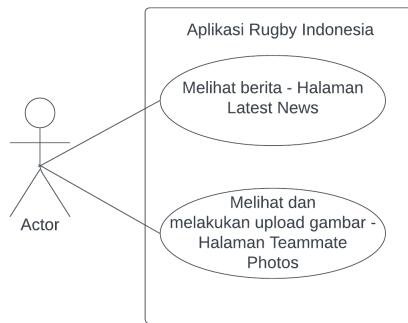
```

7 <rss xmlns:content="http://purl.org/rss/1.0/modules/content/"
8   xmlns:wfw="http://wellformedweb.org/CommentAPI/" xmlns:dc="http://purl.org/dc/elements/1.1/"
9   xmlns:atom="http://www.w3.org/2005/Atom" xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
10  xmlns:slash="http://purl.org/rss/1.0/modules/slash/" version="2.0">
11  <channel>
12    <title>Persatuan Rugby Union Indonesia</title>
13    <atom:link href="https://rugbyindonesia.or.id/feed/" rel="self" type="application/rss+xml" />
14    <link>https://rugbyindonesia.or.id</link>
15    <description>Play Rugby</description>
16    <lastBuildDate>Tue, 30 Jan 2024 06:46:29 +0000</lastBuildDate>
17    <language>en-US</language>
18    <sy:updatePeriod> hourly </sy:updatePeriod>
19    <sy:updateFrequency> 1 </sy:updateFrequency>
20    <generator>https://wordpress.org/?v=6.4.4</generator>
21    <image>
22      <url>
23        https://rugbyindonesia.or.id/wp-content/uploads/2016/03/cropped-PRUI-Default-Logo-32x32.jpg</url>
24      <title>Persatuan Rugby Union Indonesia</title>
25      <link>https://rugbyindonesia.or.id</link>
26      <width>32</width>
27      <height>32</height>
28    </image>
29    <item>
30      <title>Rugby Masuk Sekolah di Kalimantan Timur</title>
31      <link>https://rugbyindonesia.or.id/rugby-masuk-sekolah-di-kalimantan-timur/</link>
32      <dc:creator>
33        <![CDATA[ hambali ]]>
34      </dc:creator>
35      <pubDate>Tue, 30 Jan 2024 06:46:29 +0000</pubDate>
36      <category>
37        <![CDATA[ Rugby Masuk sekolah ]]>
38      </category>
39      <guid isPermaLink="false">https://rugbyindonesia.or.id/?p=6435</guid>
40      <description>
41        <![CDATA[ <!-- thumbnail:&lt;img width="820" height="490" src="https://rugbyindonesia.or.id/wp-content/uploads/2024/01/Kaltim-820x490.jpg" class="attachment-post-thumbnail size-post-thumbnail wp-post-image" alt="Kaltim-820x490.jpg" decoding="async" fetchpriority="high" /><!-- wp:paragraph --> <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> <!-- /wp:paragraph --> ]]>
42      </description>
43      <content:encoded>
44        <![CDATA[ <!-- thumbnail:&lt;img width="820" height="490" src="https://rugbyindonesia.or.id/wp-content/uploads/2024/01/Kaltim-820x490.jpg" class="attachment-post-thumbnail size-post-thumbnail wp-post-image" alt="Kaltim-820x490.jpg" decoding="async" /><!-- wp:paragraph --> <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> <!-- /wp:paragraph --> ]]]>
45      </content:encoded>
46    </item>
47  </channel>
48</rss>
```

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 berubah sehingga tidak memiliki halaman-halaman tersebut. Maka dari itu, fungsi dari aplikasi Rugby Indonesia akan berubah, dan

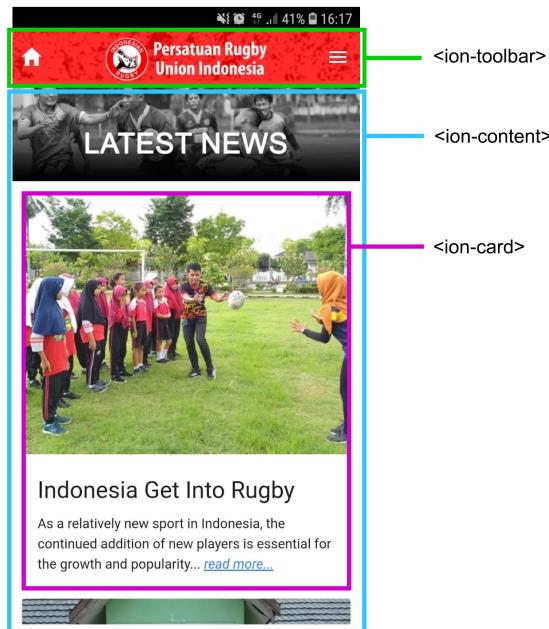
- 1 dapat dilihat seperti pada Gambar 3.3.



Gambar 3.3: *Use Case Diagram* Sistem Usulan Rugby Indonesia

2 3.4.1 Halaman Utama

- 3 Pada halaman utama, skenario dari penggunaan aplikasi Rugby Indonesia pada halaman utama
 4 sama seperti tabel skenario halaman utama dari sistem kini pada Tabel 3.1. Halaman ini memiliki
 5 beberapa komponen yang terdapat pada Ionic 7, komponen tersebut berupa Toolbar, Content, dan
 6 Card. Analisis dari komponen yang digunakan terdapat pada Gambar 3.4.



Gambar 3.4: Analisis dari Halaman Utama Rugby Indonesia

7 3.4.2 Halaman Teammate Photos

- 8 Pada halaman Teammate Photos, skenario dari aplikasi Rugby Indonesia sama dengan skenario
 9 dari halaman Teammate Photos yang sudah ada pada Tabel 3.2, namun dengan adanya sedikit
 10 penambahan, yaitu skenario di saat pengguna mengupload gambar. Penambahan ini dilakukan

- 1 karena peneliti tidak menemukan gambar di saat pengguna melakukan *upload* gambar. Penambahan
- 2 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 halaman frame
3	Pengguna memilih salah satu frame	Sistem akan mengunggah gambar ke dalam aplikasi.

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

6 Pada halaman Teammate Photos, terdapat beberapa komponen dari Ionic 7 yang digunakan.

7 Halaman ini memiliki beberapa komponen, yaitu toolbar, button, content, dan juga card. Analisis 8 dari komponen yang digunakan terdapat pada Gambar 3.5.



Gambar 3.5: Analisis dari Halaman Teammate Photos

1

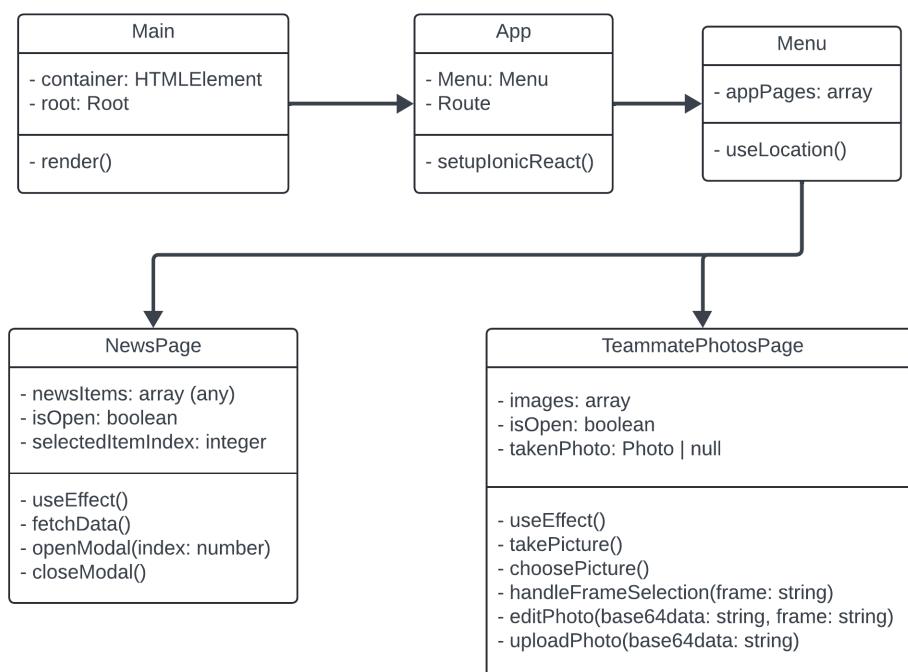
BAB 4

2

PERANCANGAN

3 4.1 Perancangan Kelas

4 Perancangan kelas diagram dilakukan setelah tahap analisis sistem kini, analisis masalah, dan
5 analisis sistem usulan yang dilakukan pada Bab 3. Dari analisis yang telah dilakukan, dihasilkan
6 sebuah diagram kelas dengan penyesuaian yang sudah dilakukan dan terdapat pada Gambar 4.1.



Gambar 4.1: Diagram Kelas Rugby Indonesia

7 Deskripsi perancangan dari tiap kelas adalah sebagai berikut:

8 1. Kelas Main

9 Kelas Main hanya bertugas untuk melakukan *render* terhadap kelas App. Pada kelas ini,
10 terdapat dua buah atribut yaitu container dan root. Selain itu, pada kelas ini juga terdapat
11 *method* render yang bertujuan untuk melakukan render terhadap App. Penjelasan terkait
12 atribut dan juga method yang terdapat pada kelas ini adalah sebagai berikut:

- 13 • Atribut container: mengambil dokumen yang memiliki element root
14 • Atribut root: membuat root dengan menggunakan createRoot yang berasal dari library

1 React.

- 2 • Method render: melakukan render pada aplikasi React.

3 2. Kelas App

4 Kelas App bertugas untuk melakukan *render* terhadap kelas Menu yang akan digunakan pada
5 aplikasi ini. Pada kelas ini, terdapat dua buah atribut yaitu Menu dan Route. Selain itu,
6 pada kelas ini juga terdapat *method* setupIonicReact. Penjelasan terkait atribut dan juga
7 *method* yang terdapat pada kelas ini adalah sebagai berikut:

- 8 • Atribut menu: memanggil kelas menu.
9 • Atribut route: melakukan *routing* terhadap *path* yang dibutuhkan.
10 • Method setupIonicReact: melakukan setup pada aplikasi dengan menggunakan IonicCon-
11 fig yang sudah disediakan dari *library*.

12 3. Kelas Menu

13 Kelas Menu berfungsi untuk melakukan pemanggilan terhadap halaman *Latest News* dan juga
14 *Teammate Photos*. Pada kelas ini, terdapat atribut appPages yang berfungsi untuk menyimpan
15 halaman dari aplikasi Rugby Indonesia. Penjelasan terkait atribut dan juga *method* yang
16 terdapat pada kelas ini adalah sebagai berikut:

- 17 • Atribut appPages: berfungsi untuk menyimpan halaman dari aplikasi yang disimpan ke
18 dalam *array*.
19 • Method render: berfungsi untuk melakukan *render* menu pada aplikasi Rugby Indonesia.

20 4. Kelas NewsPage

21 Kelas NewsPage bertugas untuk menampilkan berita-berita terkini seputar Rugby Indonesia.
22 Pada kelas ini, terdapat State newsItems, isOpen, dan juga selectedIndex. Pada kelas ini
23 juga terdapat method useEffect, fetchData, dan juga openModal. Penjelasan terkait atribut
24 state dan juga *method* yang terdapat pada kelas ini adalah sebagai berikut:

- 25 • State newsItems: berfungsi untuk menyimpan berita yang sudah diambil menggunakan
26 protokol RSS.
27 • State isOpen: berfungsi untuk menentukan apakah modal dalam keadaan terbuka atau
28 tidak.
29 • SelectedIndex: berfungsi untuk menampilkan isi dari modal yang telah dipilih.
30 • Method useEffect: melakukan pengambilan data yang berasal dari sistem eksternal.
31 • Method fetchData: melakukan pengambilan data dari url menggunakan axios.
32 • Method openModal: menampilkan isi konten menggunakan modal.
33 • Method closeModal: memberikan *default value* pada setOpen dan setSelectedIndex
34 ketika modal ditutup.

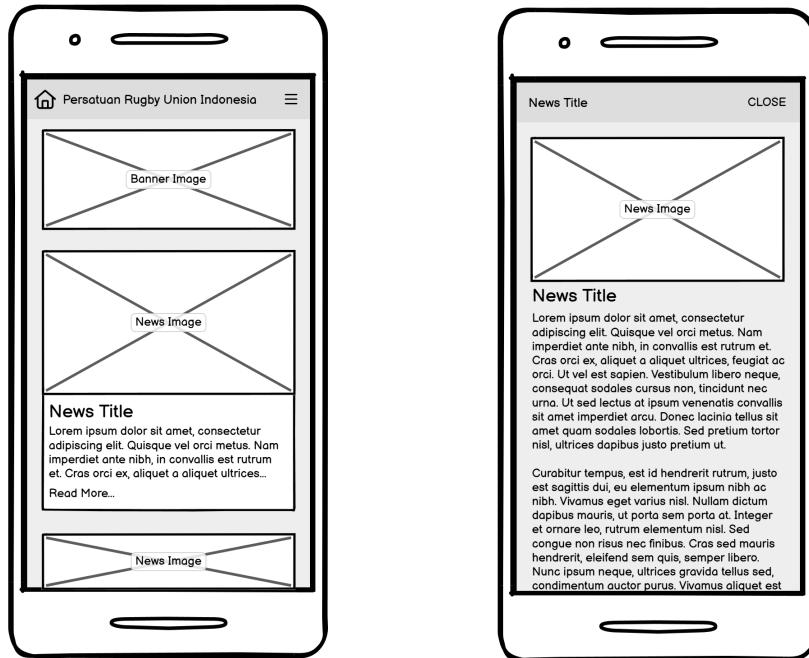
35 5. Kelas TeammatePhotosPage

36 Kelas TeammatePhotosPage bertugas untuk menampilkan foto-foto yang telah diunggah
37 oleh pengguna aplikasi Rugby Indonesia, serta melakukan *upload* foto agar foto tersebut
38 dapat dilihat oleh pengguna aplikasi Rugby Indonesia lainnya dengan cara mengambil foto
39 tersebut secara langsung atau melakukan *upload* foto dari galeri *smartphone* pengguna. Pada
40 kelas ini, terdapat State images, isOpen, dan juga takenPhoto. Pada kelas ini juga terdapat
41 method useEffect, takePicture, choosePicture, handleFrameSelection, editPhoto, uploadPhoto.
42 Penjelasan dari masing-masing atribut state dan juga method adalah sebagai berikut:

- 1 • State images: berfungsi untuk menyimpan foto yang diambil menggunakan protokol RSS.
- 2 • State isOpen: berfungsi untuk menentukan apakah modal dalam keadaan terbuka atau
- 3 tidak.
- 4 • takenPhoto: berfungsi untuk menyimpan dahulu foto sebelum pengguna melakukan
- 5 pemilihan *frame*.
- 6 • Method useEffect: melakukan pengambilan data yang berasal dari sistem eksternal.
- 7 • Method takePicture: melakukan pengambilan gambar menggunakan kamera.
- 8 • Method choosePicture: melakukan pengambilan gambar yang berasal dari galeri.
- 9 • Method handleFrameSelection: melakukan penanganan terhadap frame yang sudah
- 10 dipilih untuk dikirim ke *method* editPhoto.
- 11 • Method editPhoto: melakukan penggabungan antara foto dan juga *frame* yang telah
- 12 dipilih oleh pengguna sebelum gambar tersebut di-*upload*.
- 13 • Method uploadPhoto: melakukan pengunggahan gambar ke *server* menggunakan metode
- 14 *post*.

15 4.2 Perancangan Antarmuka

- 16 Berikut ini merupakan perancangan antarmuka dari aplikasi Rugby Indonesia untuk pengguna dari
 17 aplikasi ini.



Gambar 4.2: Perancangan Antarmuka Bagian *Latest News*

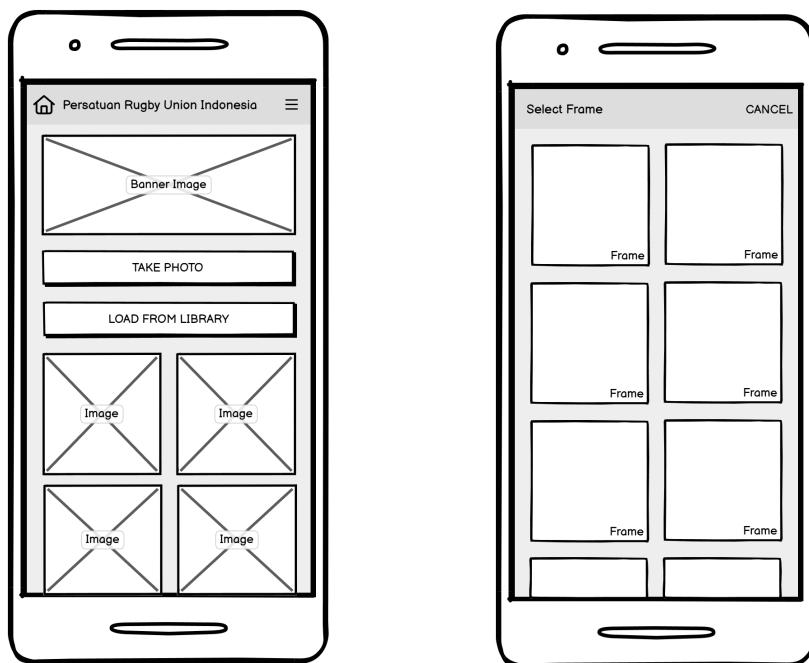
18 4.2.1 Perancangan Antarmuka Halaman *Latest News*

- 19 Pada halaman *Latest News*, informasi yang akan ditampilkan oleh halaman ini berisi berita-berita
 20 terkini seputar Rugby Indonesia. Gambar perancangan dari halaman Latest News dapat dilihat

- 1 pada Gambar 4.2a. Pada gambar tersebut, terdapat *toolbar* dari halaman *Latest News* yang berisi
 2 tombol *home*, nama dari aplikasi, dan juga tombol menu. Di bawah *toolbar*, terdapat *banner* dari
 3 halaman *Latest News*. Lalu terdapat konten yang berisi dari gambar *headline* berita, judul berita,
 4 dan deskripsi singkat dari berita tersebut. Pengguna dapat melihat deskripsi lengkap dari berita
 5 tersebut dengan melakukan klik pada tulisan “Read More...” yang terletak di bawah deskripsi
 6 singkat berita.

7 4.2.2 Perancangan Antarmuka Deskripsi Berita

- 8 Halaman ini hanya akan menampilkan detail dari berita yang ingin dibaca lebih lanjut oleh pengguna.
 9 Gambar perancangan dari halaman *Latest News* dapat dilihat pada Gambar 4.2b. Pada gambar
 10 tersebut, terdapat *toolbar* yang berisi judul berita dan juga tombol “CLOSE” untuk menutup detail
 11 dari berita. Di bawah *toolbar*, terdapat konten dari berita yang berisi gambar *headline* berita, judul
 12 berita, lalu deskripsi lengkap dari berita.



(a) Perancangan Antarmuka Halaman *Teammate Photos*

(b) Perancangan Antarmuka Halaman Pemilihan *Frame*

Gambar 4.3: Perancangan Antarmuka Bagian *Teammate Photos*

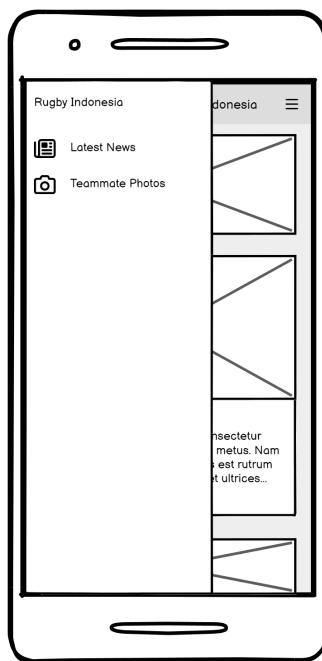
13 4.2.3 Perancangan Antarmuka Halaman *Teammate Photos*

- 14 Pada halaman *Teammate Photos*, informasi yang akan ditampilkan oleh halaman ini berisi foto-foto
 15 yang telah diunggah oleh para pengguna aplikasi Rugby Indonesia, serta tombol “TAKE PHOTO”
 16 dan “LOAD FROM LIBRARY” yang berfungsi untuk melakukan pengunggahan foto. Gambar
 17 perancangan dari halaman *Teammate Photos* dapat dilihat pada Gambar 4.3a.

¹ **4.2.4 Perancangan Antarmuka Halaman *Frame***

² Halaman ini hanya akan menampilkan pilihan *frame* yang dapat digunakan oleh pengguna Rugby
³ Indonesia setelah pengguna melakukan pengambilan gambar menggunakan kamera atau melakukan
⁴ pemilihan gambar yang berasal dari galeri. Gambar dari halaman ini dapat dilihat pada Gambar
⁵ [4.3b](#), di mana pada gambar tersebut terdapat tombol “CANCEL” apabila pengguna tidak jadi
⁶ melakukan pengunggahan gambar dan juga pilihan *frame* yang tersedia.

⁷ **4.2.5 Perancangan Antarmuka Menu**



Gambar 4.4: Perancangan Antarmuka Menu pada Aplikasi Rugby Indonesia

⁸ Perancangan ini hanya menampilkan menu dari Rugby Indonesia ketika pengguna melakukan klik
⁹ pada tombol menu di atas kanan pada aplikasi. Gambar dari perancangan ini dapat dilihat pada
¹⁰ Gambar [4.4](#). Pada gambar tersebut, terdapat dua buah pilihan menu, yaitu menu *Latest News* dan
¹¹ juga menu *Teammate Photos*.

¹² **4.3 Perancangan Struktur HTML**

¹³ Struktur HTML dari setiap halaman memiliki struktur yang sama, di mana pada setiap halaman
¹⁴ terdapat *header* dan juga *content*. *Header* dan *content* yang terdapat pada struktur ini memiliki
¹⁵ penjelasan sebagai berikut:

¹⁶ **4.3.1 *Header***

¹⁷ *Header* dari setiap halaman memiliki struktur yang sama, di mana pada setiap halaman, *header*
¹⁸ menggunakan komponen `<IonHeader>` dan memiliki komponen `<IonToolbar>`, `<IonButtons>`, dan
¹⁹ juga `<IonTitle>` di dalam header tersebut. Hal yang membedakan dari tiap *header* hanya berupa

- 1 judul yang berada di dalam komponen <IonTitle> dan juga *button* yang terdapat pada *header*
- 2 tersebut.

3 4.3.2 Content

- 4 *Content* pada setiap halaman menggunakan komponen <IonContent> di mana struktur konten
- 5 pada tiap halaman ini cukup berbeda. Struktur dari *content* sangat bergantung pada isi dari tiap
- 6 halaman. Isi dari *content* pada tiap halaman adalah sebagai berikut:

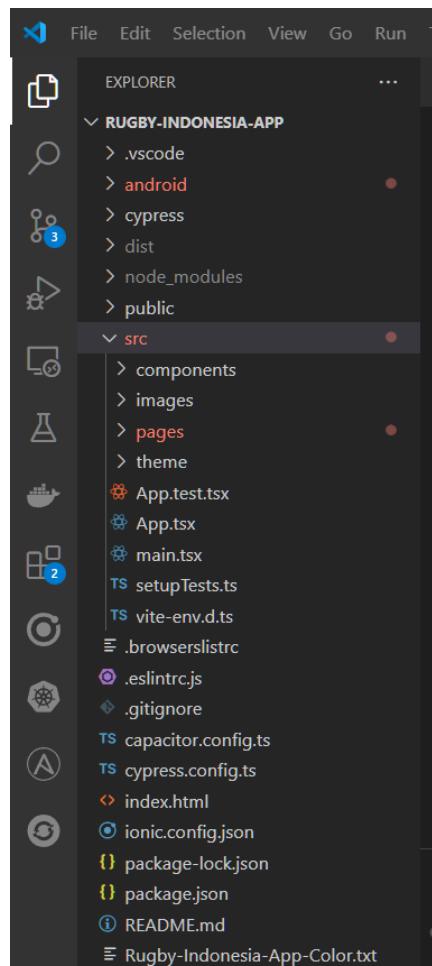
- 7 • Halaman *Latest News*

8 *Content* yang terdapat pada halaman *Latest News* berisi *banner* dari halaman *Latest News*, dan
9 juga berita terkini seputar Rugby Indonesia. Pada tiap berita yang berada di halaman *Latest*
10 *News*, berita tersebut dibungkus di dalam komponen <IonCard> yang memiliki komponen
11 <IonCardHeader> sebagai *header* dari berita dan juga komponen <IonCardContent> sebagai
12 konten dari berita tersebut.

- 13 • Halaman *Teammate Photos*

14 *Content* yang terdapat pada halaman *Teammate Photos* berisi *banner* dari halaman *Teammate*
15 *Photos*, tombol untuk melakukan pengambilan gambar menggunakan kamera, tombol untuk
16 melakukan pengunggahan gambar dari galeri, serta komponen <PhotoGallery> yang berisi
17 gambar yang telah diunggah oleh pengguna.

1 4.4 Struktur Folder dan File Aplikasi



Gambar 4.5: Struktur File Aplikasi Rugby Indonesia

- 2 Gambar 4.5 menggambarkan struktur folder dan file pada aplikasi Rugby Indonesia yang memiliki
3 penjelasan sebagai berikut:
- 4 1. src/: Direktori utama yang berisi sekumpulan berkas yang akan digunakan dalam aplikasi.
5 • components/: Direktori yang berisikan komponen dari aplikasi yang sudah disetup oleh
6 Ionic.
7 • images/: Direktori yang berisikan gambar yang akan digunakan pada aplikasi.
8 • pages/: Direktori yang berisikan tampilan halaman dari aplikasi.
9 • theme/: Direktori yang berisikan warna latar belakang yang digunakan pada aplikasi.
10 • App.tsx: Berkas yang melakukan routing antar halaman.
11 • main.tsx: Berkas yang melakukan render pada aplikasi.
12 2. package.json: Berkas konfigurasi proyek yang berisi daftar dependensi dan *script* yang perlu
13 diinstall. html: Berkas utama yang menginisialisasi aplikasi dan memuat komponen utama.

¹

BAB 5

²

IMPLEMENTASI DAN PENGUJIAN

³ 5.1 Spesifikasi Perangkat Keras

- ⁴ Pembangunan dari aplikasi Rugby Indonesia dilakukan dengan perangkat keras yang memiliki
⁵ spesifikasi sebagai berikut:
- ⁶ • Sistem Operasi: Windows 11 Home Single Language versi 23H2 64-bit
 - ⁷ • Android Development Kit: API 29 (Android 10)
 - ⁸ • Ionic: 7.0.0
 - ⁹ • Capacitor CLI: 5.4.0
 - ¹⁰ • Bahasa Pemrograman: JavaScript
 - ¹¹ • Tools: Visual Studio Code, Android Studio

¹² 5.2 Implementasi Antarmuka

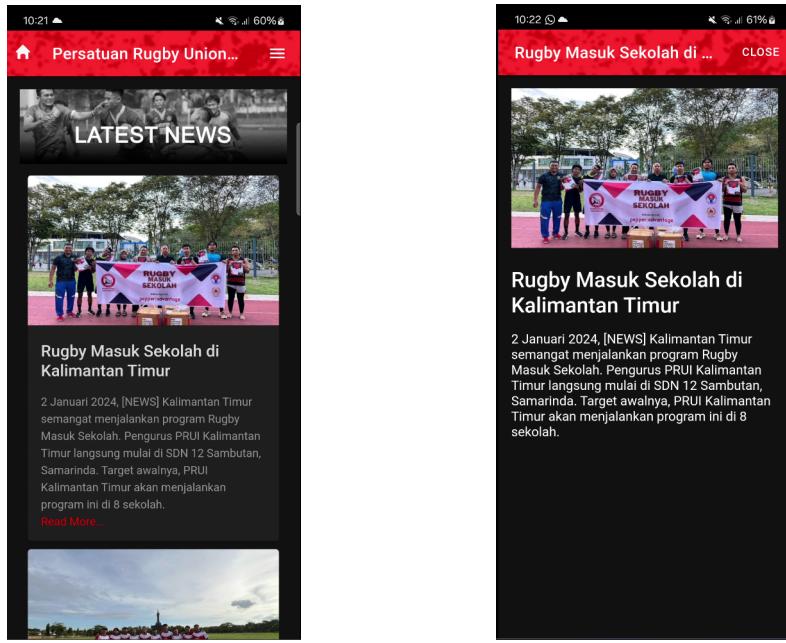
¹³ Berikut ini merupakan hasil implementasi antarmuka pada aplikasi Rugby Indonesia.

¹⁴ 5.2.1 Implementasi Antarmuka *Latest News*

¹⁵ Ketika pengguna telah mengunduh aplikasi Rugby Indonesia dan membukanya, aplikasi tersebut
¹⁶ akan menampilkan halaman *Latest News* seperti yang terdapat pada Gambar [5.1a](#). Kode yang
¹⁷ digunakan untuk melakukan implementasi ini terdapat pada Kode [A.3](#).

¹⁸ 5.2.2 Implementasi Antarmuka *News Detail*

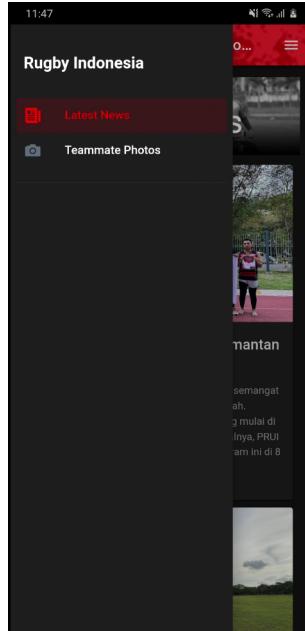
¹⁹ Halaman ini muncul ketika pengguna telah melakukan klik pada tulisan “Read More...” yang tertera
²⁰ di bawah deskripsi singkat dari masing-masing berita. Tampilan dari halaman ini terdapat pada
²¹ Gambar [5.1b](#). Kode yang digunakan untuk melakukan implementasi ini terdapat pada Kode [A.3](#)
²² baris 99 hingga 115, di mana halaman ini hanya menggunakan modal.



Gambar 5.1: Antarmuka Bagian *Latest News*

5.2.3 Implementasi Antarmuka Menu

- 1 Halaman ini ditampilkan ketika pengguna telah melakukan klik pada tombol menu di kanan atas pada halaman *Latest News* maupun *Teammate Photos*. Tampilan dari halaman ini terdapat pada Gambar 5.2. Kode yang digunakan untuk melakukan implementasi ini terdapat pada Kode A.2.



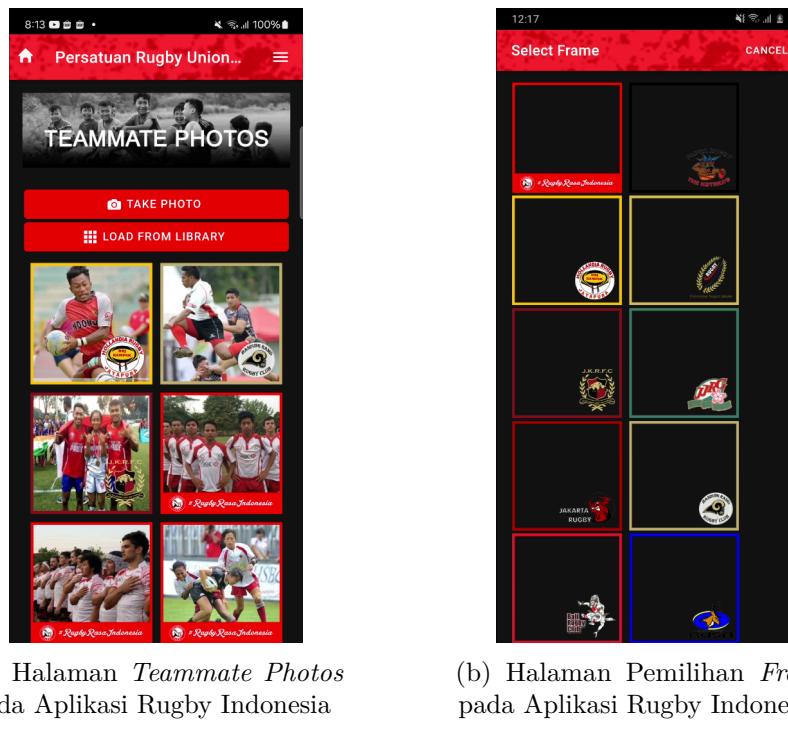
Gambar 5.2: Halaman Menu pada Aplikasi Rugby Indonesia

1 5.2.4 Implementasi Antarmuka Halaman *Teammate Photos*

2 Halaman ini ditampilkan ketika pengguna telah melakukan klik pada tombol menu di kanan atas
 3 pada halaman *Latest News* dan memilih menu *Teammate Photos*. Tampilan dari halaman ini
 4 terdapat pada Gambar 5.3a. Kode yang digunakan untuk melakukan implementasi ini terdapat
 5 pada Kode A.5.

6 5.2.5 Implementasi Antarmuka Halaman *Frame*

7 Halaman ini ditampilkan ketika pengguna telah melakukan pengambilan foto dengan melakukan
 8 klik pada tombol “TAKE PHOTO” yang terdapat pada halaman *Teammate Photos* atau telah
 9 melakukan pemilihan foto dari galeri dengan melakukan klik pada tombol “UPLOAD FROM
 10 LIBRARY” yang terdapat pada halaman *Teammate Photos* juga. Tampilan dari halaman ini
 11 terdapat pada Gambar 5.3b. Kode yang digunakan untuk melakukan implementasi ini terdapat
 12 pada Kode A.5 baris 170 hingga 196, di mana halaman ini hanya menggunakan modal.



Gambar 5.3: Antarmuka Bagian *Teammate Photos*

13 5.3 Pengujian Fungsional

14 Pada pengujian ini, dilakukan dengan beberapa tahapan yang dilakukan dengan aplikasi Rugby
 15 Indonesia. Tahapan tersebut terdapat pada Tabel 5.1 dan Tabel 5.2 berikut ini.

1 5.3.1 Pengujian Melihat Berita

Tabel 5.1: Tabel Pengujian Melihat Berita

No.	Langkah pengujian	Hasil yang diharapkan	Status
1	Menekan tombol <i>home</i> pada <i>toolbar</i>	Kembali ke awal halaman <i>Latest News</i>	Berhasil
2	Menekan tombol menu pada <i>toolbar</i>	Menampilkan menu aplikasi Rugby Indonesia	Berhasil
3	Menekan tulisan “Read More...” pada salah satu berita	Menampilkan deskripsi berita secara detail	Berhasil
4	Menekan tombol “CLOSE” pada <i>toolbar</i> di halaman <i>News Detail</i>	Kembali ke halaman <i>Latest News</i>	Berhasil

2 5.3.2 Pengujian Melihat dan Melakukan *Upload* Gambar

Tabel 5.2: Tabel Pengujian Melihat dan Melakukan *Upload* Gambar

No.	Langkah pengujian	Hasil yang diharapkan	Status
1	Menekan tombol <i>home</i> pada <i>toolbar</i>	Kembali ke halaman <i>Latest News</i>	Berhasil
2	Menekan tombol menu pada <i>toolbar</i>	Menampilkan menu dari aplikasi Rugby Indonesia	Berhasil
3	Menekan tombol “TAKE PHOTO”	Membuka kamera untuk mengambil gambar	Berhasil
4	Selesai mengambil gambar	Menampilkan halaman pemilihan <i>frame</i>	Berhasil
5	Memilih salah satu <i>frame</i>	Kembali ke halaman <i>Teammate Photos</i> dan menampilkan gambar yang telah diunggah	Berhasil
6	Menekan tombol “CANCEL” pada halaman pemilihan <i>frame</i>	Kembali ke halaman <i>Teammate Photos</i> dan tidak menampilkan gambar yang ingin diunggah	Berhasil
7	Menekan tombol “CHOOSE FROM LIBRARY”	Membuka galeri	Berhasil
8	Selesai memilih gambar dari galeri	Menampilkan halaman pemilihan <i>frame</i>	Berhasil

3 5.4 Pengujian oleh Pengguna Umum

- 4 Hasil pengujian berikut ini dilakukan oleh pengguna umum (*User Acceptance Test*) di mana pengujian ini dilakukan oleh 6 orang responden. Responden yang melakukan pengujian kali ini merupakan penggemar olahraga dengan rentang umur 21 hingga 23 tahun, jenis kelamin laki-laki, serta sedang menjalankan pendidikan S1 yang memiliki *smartphone* dengan sistem operasi Android. Responden melakukan pengujian aplikasi Rugby Indonesia dengan cara responden mengunduh aplikasi Rugby Indonesia dari tautan yang terdapat pada Google Form yang mengarahkan responden ke Google Drive. Hasil dari pengujian ini terdapat pada Tabel 5.3.

Tabel 5.3: Tabel Pengujian oleh Pengguna Umum

No.	Pertanyaan	Jawaban dari responden
1	Apakah aplikasi dapat menampilkan halaman <i>Latest News</i> saat pertama kali dibuka?	6 menjawab Ya
2	Apakah seluruh berita yang terdapat pada halaman <i>Latest News</i> ditampilkan dengan baik?	5 menjawab Ya, 1 menjawab Tidak
3	Apakah tombol "TAKE PHOTO" dapat membuka kamera <i>smartphone</i> ?	6 menjawab Ya
4	Apakah tombol "LOAD FROM LIBRARY" dapat membuka galeri <i>smartphone</i> ?	6 menjawab Ya
5	Apakah Anda mengalami kendala saat melakukan pengambilan gambar atau meng- <i>upload</i> gambar dari galeri menggunakan aplikasi Rugby Indonesia?	6 menjawab Tidak
6	Apakah Anda mengalami <i>crash</i> , <i>forced close</i> , atau kendala lain saat menggunakan aplikasi Rugby Indonesia terbaru?	6 menjawab Tidak

¹ Dari hasil yang terdapat pada Tabel 5.3, terdapat beberapa saran yang diberikan oleh responden,

² yaitu:

- ³ • Tata cara penulisan jangan terlalu panjang.
- ⁴ • Terdapat *caption* pada gambar berita.
- ⁵ • Foto yang terdapat pada halaman *Teammate Photos* dapat diperbesar.
- ⁶ • Menambahkan fitur tombol “APPLY” setelah melakukan pemilihan *frame*.

¹

BAB 6

²

KESIMPULAN DAN SARAN

³ 6.1 Kesimpulan

⁴ Berikut merupakan kesimpulan yang didapatkan dari hasil pengujian pada aplikasi Rugby Indonesia
⁵ yang telah dibangun ulang:

- ⁶ 1. Aplikasi Rugby Indonesia berhasil dibuat dengan memanfaatkan Ionic 7 dan dapat menam-
⁷ pilkan berita dengan baik.
- ⁸ 2. Aplikasi Rugby Indonesia berhasil dibuat dengan memanfaatkan Native Capacitor dan dapat
⁹ melakukan pengunggahan gambar dengan baik.

¹⁰ 6.2 Saran

¹¹ Berikut merupakan saran yang didapatkan dari hasil pengujian pada aplikasi Rugby Indonesia yang
¹² telah dibangun ulang:

- ¹³ 1. Pada halaman *Latest News*, isi konten dari berita seharusnya dapat dibuat lebih rapi lagi,
¹⁴ sehingga enak dibaca.
- ¹⁵ 2. Dapat melakukan *zoom-in* terhadap foto yang terdapat pada halaman *Teammate Photos*.
- ¹⁶ 3. Menambahkan tombol “APPLY” saat melakukan pengunggahan gambar dengan *frame*.

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] Workbench Team (2009) RSS Advisory Board. <https://www.rssboard.org/rss-specification>. [Online; diakses 7-Maret-2024].
- [4] React Axios Team (2022) react-axios. <https://www.npmjs.com/package/react-axios>. [Online; diakses 23-Maret-2024].
- [5] Connected Lab (2017) React Native XML to JSON Conversion. <https://www.npmjs.com/package/react-native-xml2js>. [Online; diakses 23-Maret-2024].
- [6] Sulistyowati (2010) *Olahraga Rugby*, 1st edition. CV Aneka Ilmu, Jl. Raya Semarang - Demak Km 8,5 Semarang.
- [7] Ionic Teams (2023) Ionic. <https://github.com/ionic-team/ionic-framework>. [Online; diakses 29-Oktober-2023].
- [8] Ionic Teams (2023) Capacitor: Cross-platform Native Runtime for Web Apps. <https://capacitorjs.com/docs>. [Online; diakses 28-November-2023].
- [9] Johnson, M. R. . M. I. P. (2014) *Product Recovery Decisions within the Context of Extended Producer Responsibility*. Journal of Engineering and Technology Management.

LAMPIRAN A

KODE PROGRAM

Kode A.1: App.tsx

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

Kode A.2: Menu.tsx

```
1 import {
2   IonContent,
3   IonIcon,
4   IonItem,
5   IonLabel,
6   IonList,
7   IonListHeader,
8   IonMenu,
9   IonMenuToggle,
10  IonNote,
11 } from '@ionic/react';
```

```

12 import { useLocation } from 'react-router-dom';
13 import { archiveOutline, archiveSharp, bookmarkOutline, cameraOutline, cameraSharp, heartOutline, heartSharp, mailOutline,
14   mailSharp, newspaper, newspaperOutline, newspaperSharp, paperPlaneOutline, paperPlaneSharp, trashOutline, trashSharp,
15   warningOutline, warningSharp } from 'ionicons/icons';
16 import './Menu.css';
17
18 interface AppPage {
19   url: string;
20   iosIcon: string;
21   mdIcon: string;
22   title: string;
23 }
24
25 const appPages: AppPage[] = [
26   {
27     title: 'Latest News',
28     url: '/latest_news',
29     iosIcon: newspaperOutline,
30     mdIcon: newspaperSharp
31   },
32   {
33     title: 'Teammate Photos',
34     url: '/teammate.photos',
35     iosIcon: cameraOutline,
36     mdIcon: cameraSharp
37 };
38
39 const Menu: React.FC = () => {
40   const location = useLocation();
41
42   return (
43     <IonMenu contentId="main" type="overlay">
44       <IonContent>
45         <IonList id="inbox-list">
46           <IonListHeader>Rugby Indonesia</IonListHeader>
47           <IonNote></IonNote>
48           {appPages.map((appPage, index) => {
49             return (
50               <IonMenuToggle key={index} autoHide={false}>
51                 <IonItem className={location.pathname === appPage.url ? 'selected' : ''} routerLink={appPage.url} routerDirection=
52                   "none" lines="none" detail={false}>
53                   <IonIcon aria-hidden="true" slot="start" ios={appPage.iosIcon} md={appPage.mdIcon} />
54                   <IonLabel>{appPage.title}</IonLabel>
55                 </IonItem>
56               </IonMenuToggle>
57             );
58           )})
59         </IonList>
60       </IonContent>
61     </IonMenu>
62   );
63 }
64
65 export default Menu;

```

Kode A.3: latest_news.tsx

```

1 import {
2   IonButtons, IonContent, IonHeader, IonMenuButton, IonPage, IonTitle, IonToolbar,
3   IonButton, IonCard, IonCardHeader, IonCardTitle, IonCardContent, IonModal, useIonViewWillLeave, useIonViewDidEnter,
4   IonBackButton, useIonViewWillEnter
5 } from '@ionic/react';
6 import { useEffect, useState } from 'react';
7 import './latest_news.css';
8
9 import bannerImage from '../../../../../images/sub-header-news.png';
10 import homeIcon from '../../../../../images/home_icon.png';
11 import axios from 'axios';
12 import xml2js from 'xml2js';
13
14 const NewsPage: React.FC = () => {
15   const [newsItems, setNewsItems] = useState([]);
16   const [isOpen, setIsOpen] = useState(false);
17   const [selectedItemIndex, setSelectedItemIndex] = useState(-1);
18
19   useEffect(() => {
20     const fetchData = async () => {
21       try {
22         const response = await axios.get('https://dnartworks.rugbyindonesia.or.id/indonesianrugby/news/list.xml');
23         const xml = response.data;
24         const result = await xml2js.parseStringPromise(xml);
25         const data = result.rss.channel[0].item;
26
27         const extractedNews = data.map((item: { description: any[]; title: any[]; 'content:encoded': any[]; }) => {
28           const extractedContent = item.description[0];
29           const regexThumbnail = /src="(.*?)"/;
30           const matchThumbnail = regexThumbnail.exec(extractedContent);
31           let getThumbnail = '';
32           if (matchThumbnail) {
33             getThumbnail = matchThumbnail[1];
34           }
35
36           let extractedTitle = item.title[0];
37           const extractedContentEncoded = item['content:encoded'][0];

```

```

38|         return {
39|             thumbnail: getThumbnail,
40|             title: extractedTitle,
41|             description: extractedContent,
42|             content: extractedContentEncoded
43|         };
44|     });
45|     setNewsItems(extractedNews);
46| } catch (error) {
47|     console.error('Error fetching data:', error);
48| }
49| );
50| );
51| fetchData();
52| [], []);
53|
54| const openModal = (index: number) => {
55|     setSelectedItemIndex(index);
56|     setIsOpen(true);
57| };
58|
59| const closeModal = () => {
60|     setIsOpen(false);
61|     setSelectedItemIndex(-1);
62| };
63|
64| return (
65|     <IonPage>
66|         <IonHeader>
67|             <IonToolbar>
68|                 <IonButtons slot="start">
69|                     <a href="/latest_news">
70|                         <img alt="home-icon" className='home-icon' />
71|                     </a>
72|                 </IonButtons>
73|                 <IonButtons slot="end">
74|                     <IonMenuButton />
75|                 </IonButtons>
76|                 <IonTitle>
77|                     Persatuan Rugby
78|                     Union Indonesia
79|                 </IonTitle>
80|             </IonToolbar>
81|         </IonHeader>
82|         <IonContent className="ion-padding">
83|             <div className="news-section-image">
84|                 <img alt='latest-news-banner' src={bannerImage} />
85|             </div>
86|
87|             {newsItems.map((item, index) => (
88|                 <IonCard key={index}>
89|                     <img alt='latest-news-image' src={item.thumbnail} />
90|                     <IonCardHeader>
91|                         <IonCardTitle>{item.title}</IonCardTitle>
92|                     </IonCardHeader>
93|                     <IonCardContent>
94|                         <div dangerouslySetInnerHTML={{ __html: item.description }}></div>
95|                         <a onClick={() => openModal(index)}>Read More...</a>
96|                     </IonCardContent>
97|                 </IonCard>
98|             )));
99|             <IonModal isOpen={isOpen} onDidDismiss={closeModal}>
100|                 <IonHeader>
101|                     <IonToolbar>
102|                         <IonTitle>{selectedItemIndex !== -1 ? newsItems[selectedItemIndex].title : ''}</IonTitle>
103|                         <IonButtons slot="end">
104|                             <IonButton onClick={closeModal}>CLOSE</IonButton>
105|                         </IonButtons>
106|                     </IonToolbar>
107|                 </IonHeader>
108|                 <IonContent className="ion-padding">
109|                     <img alt='latest-news-image' src={selectedItemIndex !== -1 ? newsItems[selectedItemIndex].thumbnail : ''} />
110|                     <h1>{selectedItemIndex !== -1 ? newsItems[selectedItemIndex].title : ''}</h1>
111|                     {selectedItemIndex !== -1 && (
112|                         <div dangerouslySetInnerHTML={{ __html: newsItems[selectedItemIndex].content }}></div>
113|                     )}
114|                 </IonContent>
115|             </IonModal>
116|         </IonContent>
117|     </IonPage>
118| );
119| );
120| export default NewsPage;

```

Kode A.4: latest_news.css

```

1 .home-icon{
2     width: calc(32rem / 16);
3     height: auto;
4 }
5
6 figure img{
7     height: 100%;
8     width: 100%;
9 }

```

```

10 ion-toolbar {
11   --background: url('../images/header.png');
12   --color: white;
13 }

```

Kode A.5: teammate_photos.tsx

```

1 import React, { useEffect, useState } from 'react';
2 import { IonButtons, IonContent, IonHeader, IonMenuButton, IonPage, IonTitle, IonToolbar, IonButton, IonIcon, IonModal, IonCol } from '@ionic/react';
3 import { Camera, Photo, CameraResultType, CameraSource } from '@capacitor/camera';
4
5 import './teammate_photos.css';
6
7 import bannerImage from '../images/sub-header-photo.png';
8 import homeIcon from '../images/home.icon.png';
9
10 import { PhotoImages } from "./photoImages";
11 import PhotoGallery from './photoGallery';
12 import { appSharp, camera } from 'ionicons/icons';
13
14 import frame1 from '../images/frame/frame01.png';
15 import frame2 from '../images/frame/frame02.png';
16 import frame3 from '../images/frame/frame03.png';
17 import frame4 from '../images/frame/frame04.png';
18 import frame5 from '../images/frame/frame05.png';
19 import frame6 from '../images/frame/frame06.png';
20 import frame7 from '../images/frame/frame07.png';
21 import frame8 from '../images/frame/frame08.png';
22 import frame9 from '../images/frame/frame09.png';
23 import frame10 from '../images/frame/frame10.png';
24
25 const frames = [frame1, frame2, frame3, frame4, frame5, frame6, frame7, frame8, frame9, frame10];
26
27 const TeammatePhotosPage: React.FC = () => {
28   const [images, setImages] = useState<PhotoImages[]>([]);
29   const [isOpen, setIsOpen] = useState(false);
30   const [takenPhoto, setTakenPhoto] = useState<Photo | null>(null);
31
32   useEffect(() => {
33     const fetchData = async () => {
34       try {
35         const response = await fetch('https://dnartworks.rugbyindonesia.or.id/indonesianrugby/photos/list.json');
36         const data = await response.json();
37         const result = data.data;
38
39         if (!Array.isArray(result)) {
40           throw new Error('Data is not in expected format');
41         }
42
43         const photoUrls = result.map((item: any) => {
44           const photoUrl = item;
45           const modPhotoUrl = photoUrl.replace("/images/", "/");
46           return modPhotoUrl;
47         });
48         const photoImages = photoUrls.map((url: string) => ({ webviewPath: url }));
49         setImages(photoImages);
50       } catch (error) {
51         console.error('Error fetching data:', error);
52       }
53     };
54
55     fetchData();
56   }, []);
57
58   const takePicture = async () => {
59     const image = await Camera.getPhoto({
60       quality: 100,
61       allowEditing: false,
62       resultType: CameraResultType.Base64,
63       source: CameraSource.Camera
64     });
65
66     setTakenPhoto(image);
67     setIsOpen(true); // Buka modal setelah foto diambil
68   };
69
70   const choosePicture = async () => {
71     const image = await Camera.getPhoto({
72       quality: 100,
73       allowEditing: false,
74       resultType: CameraResultType.Base64,
75       source: CameraSource.Photos
76     });
77
78     setTakenPhoto(image);
79     setIsOpen(true); // Buka modal setelah foto dipilih
80   };
81
82   const editPhoto = async (base64data: string, frame: string): Promise<string> => {
83     return new Promise((resolve) => {
84       const canvas = document.createElement('canvas');
85       const ctx = canvas.getContext('2d');
86       const img = new Image();
87       const frameImg = new Image();
88

```

```

89         img.onload = () => {
90             canvas.width = 400;
91             canvas.height = 400;
92             ctx!.drawImage(img, 0, 0, canvas.width, canvas.height);
93             frameImg.src = frame;
94             frameImg.onload = () => {
95                 ctx!.drawImage(frameImg, 0, 0, canvas.width, canvas.height);
96                 resolve(canvas.toDataURL('image/jpeg'));
97             };
98         };
99     );
100    img.src = 'data:image/jpeg;base64,' + base64data;
101  });
102};

103 const uploadPhoto = async (base64data: string) => {
104   try {
105     const userId = "anonymous";
106
107     const response = await fetch('https://dnartworks.rugbyindonesia.or.id/indonesianrugby/photos/upload.json', {
108       method: 'POST',
109       headers: {
110         'Content-Type': 'application/x-www-form-urlencoded'
111       },
112       body: `userId=${userId}&photo=${base64data}`
113     });
114
115     if (response.ok) {
116       console.log('Photo uploaded successfully!');
117     } else {
118       console.error(`Failed to upload photo: ${response.statusText}`);
119     }
120   } catch (error) {
121     console.error(`Error uploading photo: ${error}`);
122   }
123 };
124};

125 const handleFrameSelection = async (frame: string) => {
126   if (takenPhoto) {
127     const editedBase64Data = await editPhoto(takenPhoto.base64String!, frame);
128     await uploadPhoto(editedBase64Data);
129     window.location.reload();
130   }
131   setIsOpen(false);
132 };
133};

134 return (
135   <IonPage>
136     <IonHeader>
137       <IonToolbar>
138         <IonButtons slot="start">
139           <a href="/latest_news">
140             <img src={homeIcon} alt="home-icon" className='home-icon' />
141           </a>
142         </IonButtons>
143         <IonButtons slot="end">
144           <IonMenuButton />
145         </IonButtons>
146       <IonTitle>
147         Persatuan Rugby
148         Union Indonesia
149       </IonTitle>
150     </IonToolbar>
151   </IonHeader>
152   <IonContent className="ion-padding">
153     <div className='teammate-photos-banner'>
154       <img src={bannerImage} alt="Teammate-Photos-Banner" />
155     </div>
156     <br/><br/>
157     <IonButton color="primary" expand="block" onClick={takePicture}>
158       <IonIcon slot="start" icon={camera}></IonIcon>
159       Take Photo
160     </IonButton>
161
162     <IonButton color='primary' expand='block' onClick={choosePicture}>
163       <IonIcon slot="start" icon={appsSharp}></IonIcon>
164       Load from Library
165     </IonButton>
166
167     <PhotoGallery photos={images} />
168
169     <IonModal isOpen={isOpen} onDidDismiss={() => setIsOpen(false)}>
170       <IonHeader>
171         <IonToolbar>
172           <IonTitle>Select Frame</IonTitle>
173           <IonButtons slot="end">
174             <IonButton onClick={() => setIsOpen(false)}>CANCEL</IonButton>
175           </IonButtons>
176         </IonToolbar>
177       </IonHeader>
178       <IonContent className="ion-padding">
179         <div className="frame-selection">
180           {frames.map((frame, index) => (
181             <IonCol size = "6" key={index}>
182               <img
183                 key={index}
184                 src={frame}
185                 alt={'Frame ${index + 1}'}
186                 className="frame-option"
187             </IonCol>
188           ))}
189         </IonContent>
190     </IonModal>
191   </IonContent>
192 
```

```

188         onClick={() => handleFrameSelection(frame)}
189         width={150}
190         height={150}
191     />
192   </IonCol>
193   ))}
194   </div>
195   </IonContent>
196   </IonModal>
197   </IonContent>
198   </IonPage>
199 );
200 };
201
202 export default TeammatePhotosPage;

```

Kode A.6: teammate_photos.css

```

1 .home-icon{
2   width: calc(32rem / 16);
3   height: auto;
4 }
5
6 ion-toolbar {
7   --background: url('.../images/header.png');
8   --color: white;
9 }

```

Kode A.7: photoGallery.tsx

```

1 import { IonCol, IonGrid, IonImg, IonRow } from "@ionic/react";
2 import { PhotoImages } from "./photoImages";
3 import React from "react";
4
5 type Props = {
6   photos: PhotoImages[],
7 }
8
9 const PhotoGallery: React.FC<Props> =({photos}) => {
10   return (
11     <IonGrid>
12       <IonRow>
13         {photos.map((photo, idx) =>(
14           <IonCol size = "6" key={idx}>
15             <IonImg src={photo.webviewPath}/>
16           </IonCol>
17         )));
18       </IonRow>
19     </IonGrid>
20   );
21 }
22
23 export default PhotoGallery;

```

Kode A.8: photoImages.tsx

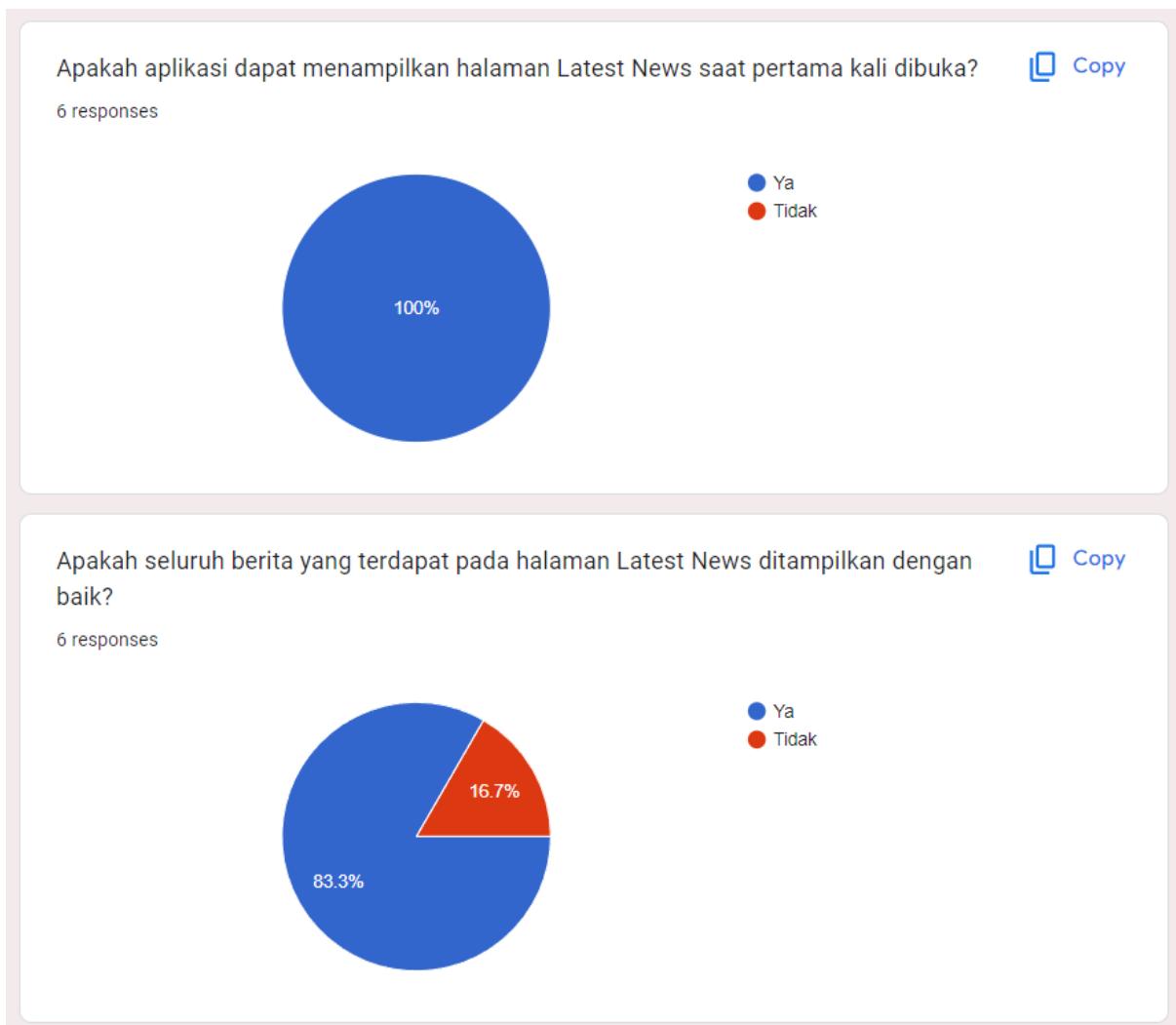
```

1 export interface PhotoImages
2 {
3   filePath: string;
4   webviewPath?: string;
5 }

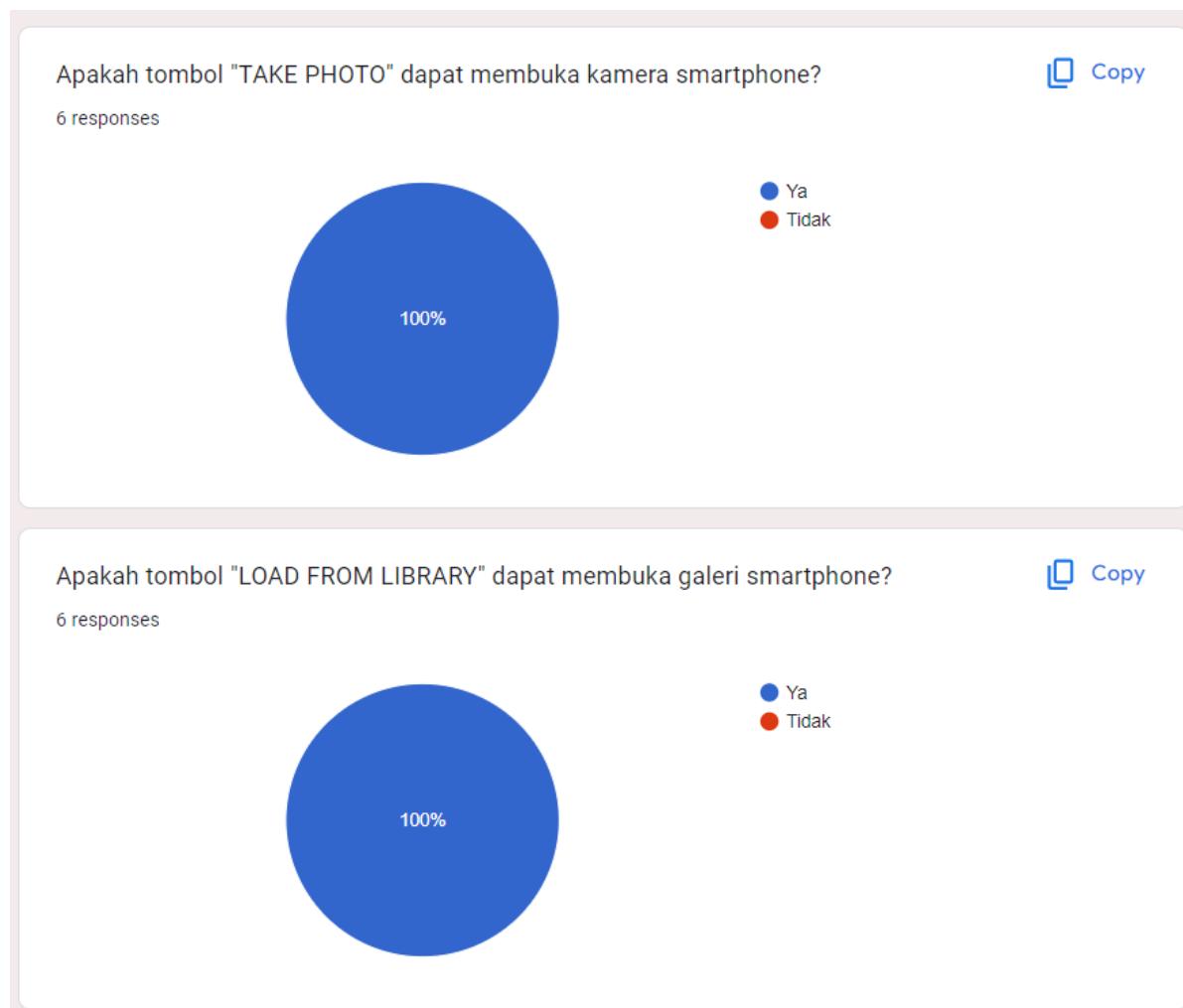
```

LAMPIRAN B

HASIL KUESIONER ACCEPTANCE TEST



Gambar B.1: Tangkapan Layar Pertama Hasil Kuesioner UAT



Gambar B.2: Tangkapan Layar Kedua Hasil Kuesioner UAT



Gambar B.3: Tangkapan Layar Ketiga Hasil Kuesioner UAT

Apakah Anda memiliki kritik dan saran terhadap aplikasi Rugby Indonesia?

6 responses

Untuk aplikasinya sudah lumayan baik, masih ada beberapa bagian yang dapat diperbaiki seperti tata cara penulisan beritanya jangan terlalu panjang, kalau mau panjang bisa dipisah dengan center / gambar, usahakan ada caption pada gambar untuk beritanya, dan untuk bagian teammate foto saya tidak dapat zoom foto-foto yang ada digalerinya, dan buat pilih frame lalu upload nya ada delay sehingga terupload nya banyak, dan apakah bisa diadakan fitur untuk hapus foto galeri yang telah diupload?

menyediakan mode terang juga

Sudah good

Tampilan antarmuka mudah digunakan dan dipahami

-

Aplikasi menurut saya sudah optimal bahkan di sistem android lawas, namun jika dibuka oleh emulator android kadang terjadi force close saat membuka galeri tapi tidak signifikan terjadi.

Gambar B.4: Tangkapan Layar Keempat Hasil Kuesioner UAT