

SKRIPSI

PEMBUATAN ULANG APLIKASI RUGBY INDONESIA DENGAN IONIC 7 DAN CAPACITOR



Nathanael Adi Trianto

NPM: 6181901041

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
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 atau yang biasa disingkat dengan PRUI merupakan Asosiasi Uni Rugbi Indonesia yang bertanggung jawab atas pengelolaan dan pengembangan Rugby Union di Indonesia yang dibangun. Rugby Indonesia juga memiliki *website*, di mana *website* tersebut didirikan oleh PT. Dnartworks Komunikasi Visual untuk memberikan info seputar kegiatan olahraga *rugby* yang berada di Indonesia. Pada sekitar tahun 2015, perusahaan PT. DNArtworks Komunikasi Visual membuat aplikasi Rugby Indonesia yang memanfaatkan Apache Cordova, namun pada saat ini aplikasi tersebut tidak dapat dipasang pada perangkat Android dikarenakan wensite yang sudah berubah dan Apache Cordova yang tidak digunakan kembali pada pengembangan aplikasi berbasis framework Ionic dikarenakan Ionic sendiri telah mengembangkan Native Capacitor sehingga Ionic dapat melepas ketergantungannya pada Angular dan Cordova.

Aplikasi Rugby Indonesia saat ini dibangun dengan menggunakan Ionic 7, diimplementasikan dengan menggunakan *library* Javascript dari React. Pada *framework* Ionic, terdapat UI Component yang dapat 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 *icon* 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 mudah untuk menggunakan aplikasi yang dibuat. Salah satu contoh dari Capacitor Native ini yaitu penggunaan Capacitor Camera yang dapat melakukan penangkapan gambar. Maka dari itu, 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 dengan 2 tahap, yaitu pengujian fungsionalitas dan pengujian oleh pengguna umum yang dilakukan oleh 6 responden dengan cara mengunduh aplikasi tersebut dari tautan Google Drive yang telah dibagikan lalu responden diminta untuk mengisi kuesioner yang diberikan. Hasil yang didapatkan dari pengujian tersebut yaitu aplikasi berjalan dengan lancar pada perangkat Android.

Kata-kata kunci: Rugby Indonesia, *framework* Ionic, UI Component, Capacitor, React, Android

ABSTRACT

Persatuan Rugby Union Indonesia or commonly abbreviated as PRUI is the Indonesian Rugby Union Association responsible for the management and development of Rugby Union in Indonesia. Rugby Indonesia also has a website, which was established by PT Dnartworks Komunikasi Visual to provide information about rugby activities in Indonesia. Around 2015, the company PT DNArtworks Komunikasi Visual created the Rugby Indonesia application that utilizes Apache Cordova, but at this time the application cannot be installed on Android devices because the website has changed and Apache Cordova is not reused in the development of applications based on the Ionic framework because Ionic itself has developed Native Capacitor so that Ionic can release its dependence on Angular and Cordova.

The Rugby Indonesia application is currently built using Ionic 7, implemented using the Javascript library from React. In the Ionic framework, there are UI Components that can allow 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 the functionality of the original device 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. Therefore, 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, but the tests carried out in this final project were only tested on Android devices.

Application testing is carried out in 2 stages, namely functionality testing and testing by general users conducted by 6 respondents by downloading the application from the Google Drive link that has been distributed and then the respondents are asked to fill out the questionnaire given. The results obtained from the test are that the application runs smoothly on Android devices.

Keywords: Indonesia Rugby, Ionic Framework, UI Component, 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	4
1.4 Batasan Masalah	4
1.5 Metodologi	4
2 LANDASAN TEORI	5
2.1 Aplikasi Rugby Indonesia	5
2.2 ReactJS [1]	5
2.2.1 Hooks	6
2.2.2 Components	7
2.2.3 Dangerously setting the inner HTML	9
2.3 Ionic 7 Framework [2]	9
2.3.1 UI Components	9
2.3.2 Capacitor Native	17
2.4 RSS [3]	19
2.5 React-Axios [4]	19
2.6 React Native XML to JSON Conversion [5]	20
3 ANALISIS	21
3.1 Analisis Sistem Kini	21
3.1.1 Halaman Utama	22
3.1.2 Halaman Teammate Photos	22
3.1.3 Halaman Rugby Clubs	22
3.1.4 Halaman Fixture & Results	23
3.2 Analisis Masalah Aplikasi Rugby Indonesia	23
3.3 Analisis RSS Rugby Indonesia	24
3.4 Analisis Sistem Usulan	24
3.4.1 Halaman Utama	25
3.4.2 Halaman Teammate Photos	25
4 PERANCANGAN	27
4.1 Perancangan Kelas	27
4.2 Perancangan Antarmuka	29
4.2.1 Perancangan Antarmuka Halaman Latest News	29
4.2.2 Perancangan Antarmuka Deskripsi Berita	30
4.2.3 Perancangan Antarmuka Halaman Teammate Photos	30
4.2.4 Perancangan Antarmuka Halaman Frame	30

4.3	Perancangan Struktur HTML	31
4.3.1	Header	31
4.3.2	Content	31
5	IMPLEMENTASI DAN PENGUJIAN	33
5.1	Spesifikasi Perangkat Lunak	33
5.2	Implementasi Antarmuka	33
5.2.1	Implementasi Antarmuka News Page	33
5.2.2	Implementasi Antarmuka News Detail	34
5.2.3	Implementasi Antarmuka Teammate Photos Page	34
5.2.4	Implementasi Antarmuka Halaman Frame	35
5.3	Pengujian Fungsional	36
5.3.1	Pengujian Melihat Berita	36
5.3.2	Pengujian Melihat dan Melakukan Upload Gambar	36
5.4	Pengujian oleh Pengguna Umum	36
6	KESIMPULAN DAN SARAN	39
6.1	Kesimpulan	39
6.2	Saran	39
DAFTAR REFERENSI		41
A	KODE PROGRAM	43

DAFTAR GAMBAR

1.1 Halaman aplikasi Rugby Indonesia	3
2.1 Gambar Hasil Action Sheet	10
3.1 Use Case Diagram Rugby Indonesia App	21
3.2 Halaman Utama Aplikasi Rugby Indonesia	23
3.3 UCD Sistem Usulan Rugby Indonesia	25
3.4 Analisis dari Halaman Utama Rugby Indonesia	25
3.5 Analisis dari Halaman Teammate Photos	26
4.1 Diagram Kelas Rugby Indonesia	27
4.2 Perancangan Antarmuka Bagian Latest News	29
4.3 Perancangan Antarmuka Bagian Teammate Photos	30
5.1 Halaman Latest News pada Aplikasi Rugby Indonesia	34
5.2 Halaman Deskripsi Berita pada Aplikasi Rugby Indonesia	34
5.3 Halaman Teammate Photos pada Aplikasi Rugby Indonesia	35
5.4 Halaman Pemilihan Frame pada Aplikasi Rugby Indonesia	35

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 juga pertama kali
8 diperkenalkan pada tahun 1986 pada zaman penjajahan, dan dihidupkan kembali pada tahun 2004
9 dengan bantuan negara lain seperti Australia dan Selandia Baru sehingga terbentuklah Indonesian
10 Rugby Football Union (IRFU) yang bertujuan untuk mengembangkan olahraga ini.

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

22 Dikutip dari Dicoding⁴, Wordpress merupakan sebuah platform Content Management System
23 (CMS) yang sangat populer untuk membangun *website* dan blog. Diluncurkan pada tahun 2004
24 oleh Matt Mullenweg dan Mike Little, WordPress memungkinkan pengguna untuk membuat dan
25 mengelola *website* tanpa perlu memiliki pengetahuan dasar pemrograman. Content Management
26 System (CMS) yang dibangun WordPress menggunakan bahasa pemrograman PHP dan database
27 MySQL. Dengan WordPress, pengguna baru dapat mengelola website tanpa perlu memahami teknis
28 bahasa pemrograman, hal itu pun yang membuat para pemula dapat dengan mudah membuat
29 website, bahkan dalam hitungan menit.

30 Dikutip dari GitHub Ionic [7], Ionic Framework merupakan *toolkit UI open-source* untuk
31 membangun aplikasi modern, *cross-platform* yang berkualitas tinggi dari satu kode sumber dengan

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

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

³<https://rugbyindonesia.or.id/>

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

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

17 Dikutip dari dokumentasi Capacitor [8], Capacitor merupakan *runtime native open source* yang
18 dapat digunakan untuk membangun aplikasi Web Native. Capacitor memungkinkan pengembang
19 untuk membangun aplikasi *mobile* pada perangkat iOS dan Android. Selain itu, Capacitor juga
20 memungkinkan pengembang untuk membangun aplikasi web progresif. Aplikasi yang dibangun
21 akan menggunakan JavaScript, HTML, dan CSS. Capacitor juga menyediakan *cross-platform API*
22 yang dapat digunakan pada berbagai platform seperti iOS, Android, Web dengan satu kode sumber.
23 Capacitor ini sendiri dikembangkan oleh tim Ionic, di mana tim ini jugalah yang mengembangkan
24 *framework* Ionic. Capacitor dirancang untuk menggantikan Apache Cordova, yang merupakan
25 *runtime* yang digunakan oleh Ionic sebelumnya. Capacitor ini juga memiliki beberapa keunggulan
26 dibandingkan Apache Cordova, diantaranya adalah lebih ringan, lebih cepat, lebih modern, dan
27 lebih fleksibel.

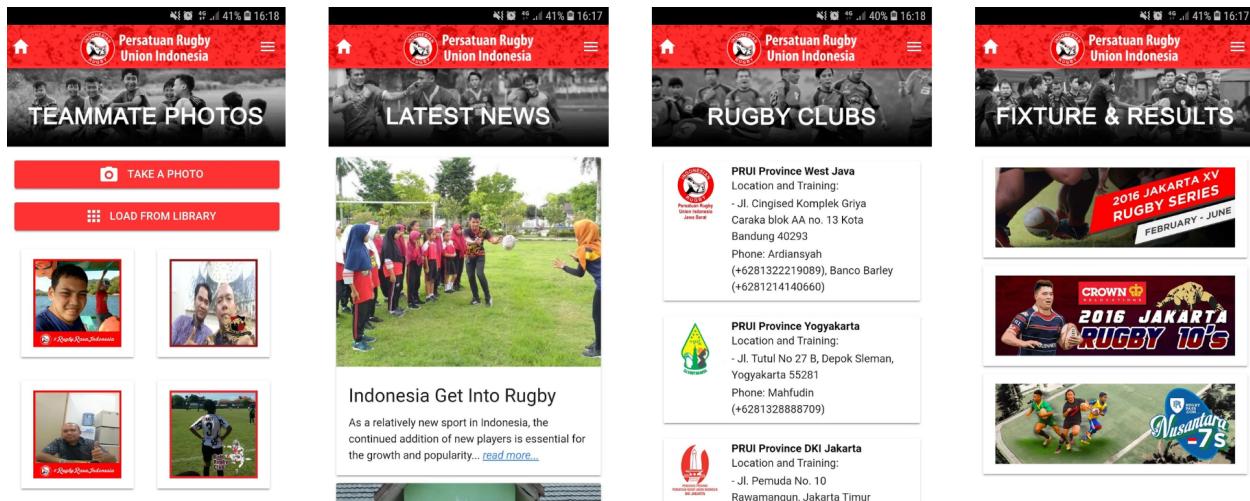
28 Dikutip dari dokumentasi RSS⁵, RSS atau Really Simple Syndication adalah format sindikasi
29 konten web yang merupakan dialek dari XML. Semua file RSS harus disesuaikan dengan spesifikasi
30 XML 1.0 yang dipublikasikan oleh World Wide Web Consortium (W3C). Dokumen RSS terdiri
31 dari elemen <*rss*>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

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

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

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

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



Gambar 1.1: Halaman aplikasi Rugby Indonesia dari Google Play Store

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

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

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

20 1.2 Rumusan Masalah

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

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

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

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

1.3 Tujuan

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

6 1.4 Batasan Masalah

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

15 1.5 Metodologi

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

¹

BAB 2

²

LANDASAN TEORI

³ 2.1 Aplikasi Rugby Indonesia

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

¹⁴ Fitur-fitur yang ada pada aplikasi Rugby Indonesia dengan versi terakhir yang ada saat ini
¹⁵ yaitu:

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

²⁶ 2.2 ReactJS [1]

²⁷ ReactJS atau React adalah sebuah *library* JavaScript yang digunakan untuk membangun *user*
²⁸ *interface* yang interaktif. ReactJS berisi kumpulan potongan kode JavaScript yang disebut “kompo-
²⁹ nen” yang bisa digunakan berulang kali untuk mendesain antarmuka pengguna. ReactJS bukanlah
³⁰ *framework* JavaScript, karena hanya bertugas untuk me-*render* komponen area tampilan aplikasi.
³¹ ReactJS dapat digunakan untuk membuat aplikasi web dan *mobile*.

1 2.2.1 Hooks

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

5 State Hooks

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

Kode 2.1: Contoh Potongan Kode State Hooks

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

15 Context Hooks

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

Kode 2.2: Contoh Potongan Kode Context Hooks

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

25 Ref Hooks

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

Kode 2.3: Contoh Potongan Kode Ref Hooks

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

36 Effect Hooks

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

Kode 2.4: Contoh Potongan Kode Effect Hooks

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

10 Performance Hooks

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

Kode 2.5: Contoh Potongan Kode Performance Hooks

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

21 Resource Hooks

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

Kode 2.6: Contoh Potongan Kode Resource Hooks

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

31 2.2.2 Components

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

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

41 Fragment

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

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

Kode 2.7: Contoh Potongan Kode Fragment

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

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

18 Profiler

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

Kode 2.8: Contoh Potongan Kode Profiler

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

27 Suspense

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

Kode 2.9: Contoh Potongan Kode Suspense

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

37 StrictMode

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

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

Kode 2.10: Contoh Potongan Kode StrictMode

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

¹⁷ 2.2.3 Dangerously setting the inner HTML

18 dangerouslySetInnerHTML merupakan sebuah objek dengan bentuk {__html:'<p>some html</p>'}
19 yang berisi string HTML mentah di dalamnya. Hal ini akan menimpa properti innerHTML dari
20 node DOM dan menampilkan HTML yang diberikan di dalamnya. Contoh kode penggunaan
21 dangerouslySetInnerHTML terdapat pada Kode 2.11. Pada kode tersebut, pengembang harus
22 memastikan bahwa kode HTML tersebut aman atau berasal dari sumber yang terpercaya.

Kode 2.11: Contoh Potongan Kode dangerouslySetInnerHTML

```
23
24 1 const markup = { __html: '<p>some raw html</p>' };
25 2 return <div dangerouslySetInnerHTML={markup} />;
```

27 2.3 Ionic 7 Framework [2]

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

32 2.3.1 UI Components

³³ 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:

37 Action Sheet

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

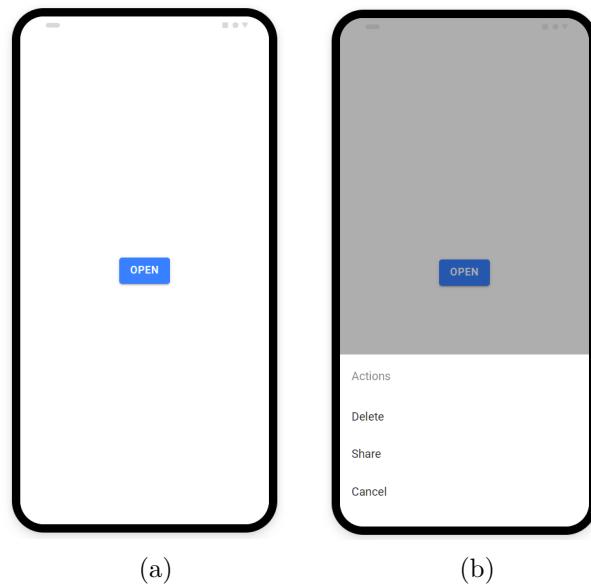
Kode 2.12: Contoh Potongan Kode Action Sheet

```

1 import React from 'react';
2 import { IonActionSheet, IonButton } from '@ionic/react';
3
4 function Example() {
5   return (
6     <>
7       <IonButton id="open-action-sheet">Open</IonButton>
8       <IonActionSheet
9         trigger="open-action-sheet"
10        header="Actions"
11        buttons={[
12          {
13            text: 'Delete',
14            role: 'destructive',
15            data: {
16              action: 'delete',
17            },
18          },
19          {
20            text: 'Share',
21            data: {
22              action: 'share',
23            },
24          },
25          {
26            text: 'Cancel',
27            role: 'cancel',
28            data: {
29              action: 'cancel',
30            },
31          },
32        ]}
33      ></IonActionSheet>
34    </>
35  );
36}
37
38 export default Example;

```

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



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

41 Properti ‘role’ pada Action Sheet adalah sebuah properti yang diberikan untuk memberikan *role*
42 ‘cancel’ atau ‘destructive’ pada tombol yang berada di dalam Action Sheet. Nilai ‘cancel’ digunakan
43 untuk tombol yang akan membatalkan aksi yang dilakukan, sedangkan nilai ‘destructive’ digunakan

- 1 untuk tombol yang akan menghapus atau mengubah data yang ada. Selain properti role, Action
 2 Sheet memiliki properti seperti pada (Tabel 2.1).

Tabel 2.1: Tabel Properti dari Action Sheet

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

3 Button

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

Kode 2.13: Contoh Potongan Kode Button

```
6 1 <IonButton>Open</IonButton>
```

9 Pada komponen Button terdapat properti 'expand' dan 'icons'. Secara *default*, komponen
 10 Button memiliki *style display*: *inline-block* dan tidak memiliki ikon. Namun dengan properti
 11 *expand*, *style* pada komponen Button dapat diubah dengan cara memberikan properti 'expand' pada
 12 komponen Button seperti `<IonButton expand="block">Block</IonButton>`. Properti ikon juga
 13 dapat ditambahkan pada awal, akhir, ataupun hanya terdapat ikon pada Button tersebut, seperti
 14 pada (Kode 2.14). Saat menambahkan Icon, pengembang bisa mengikuti cara yang terdapat pada
 15 subbab 2.3.1.

Kode 2.14: Contoh Potongan Kode Button Menggunakan Icon

```
16 1 <IonButton>
17 2   <IonIcon slot="start" icon={star}></IonIcon>
18 3   Left Icon
19 4 </IonButton>
```

22 Nilai dari 'slot' dapat berupa 'start' untuk menempatkan ikon di awal Button, 'end' untuk
 23 menempatkan ikon di akhir button, ataupun 'icon-only' untuk memberikan ikon saja pada button.

24 Card

25 Card merupakan komponen yang digunakan untuk menampilkan konten seperti teks, gambar,
 26 tombol, dan daftar dalam sebuah kotak. Komponen ini biasanya terdiri dari *header*, judul, gambar,
 27 dan konten utama. Card dapat digunakan sebagai komponen tunggal atau digabungkan dengan
 28 komponen lain untuk membuat tampilan yang lebih kompleks. Card dapat disesuaikan dengan
 29 menggunakan properti CSS seperti 'background' dan 'color'. Cara menggunakan Card terdapat
 30 pada contoh (Kode 2.15).

Kode 2.15: Contoh Potongan Kode Card

```
31 1 import React from 'react';
32 2 import { IonCard, IonCardContent, IonCardHeader, IonCardSubtitle, IonCardTitle } from '@ionic/react';
33 3
34 4 function Example() {
```

```

15   return (
16     <IonCard>
17       <IonCardHeader>
18         <IonCardTitle>Card Title</IonCardTitle>
19         <IonCardSubtitle>Card Subtitle</IonCardSubtitle>
20       </IonCardHeader>
21
22       <IonCardContent>Here's a small text description for the card content. Nothing more, nothing less.</IonCardContent>
23     </IonCard>
24   );
25 }
26 export default Example;

```

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

17 Content

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

Kode 2.16: Contoh Potongan Kode Content

```

23 1 import React from 'react';
23 2 import { IonContent } from '@ionic/react';
23 3
24 function Example() {
25   return (
26     <IonContent className="ion-padding">
27       <h1>Heading 1</h1>
28
29       <p>Here's a small text description for the content. Nothing more, nothing less.</p>
30     </IonContent>
31   );
32 }
33 export default Example;

```

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

40 Grid

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

46 Icon

Komponen Icon adalah elemen dasar yang tersedia melalui *library* Ionicons. Secara *default*, Icon dipersiapkan dengan semua aplikasi Ionic Framework. Komponen ini dapat digunakan untuk menampilkan ikon dari set Ionicons atau ikon yang berupa SVG. Selain itu, komponen ini juga 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;

```

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

21 List

22 List digunakan untuk menampilkan data dalam bentuk baris, seperti daftar kontak, daftar putar, 23 atau menu. Komponen ini mendukung berbagai macam interaksi, termasuk menggeser *item* untuk 24 menampilkan opsi, menarik untuk menyusun ulang *item* dalam daftar, dan menghapus *item*. 25 Komponen ini dapat ditambah berbagai elemen ke dalam daftar, seperti teks, tombol, ikon, dan 26 gambar dengan ukuran yang kecil. Cara penggunaan List terdapat pada (Kode 2.18).

Kode 2.18: Contoh Kode Penggunaan List

```

27 import React from 'react';
28 import { IonItem, IonLabel, IonList } from '@ionic/react';
29
30
31 function Example() {
32   return (
33     <IonList>
34       <IonItem>
35         <IonLabel>Pokemon Yellow</IonLabel>
36       </IonItem>
37       <IonItem>
38         <IonLabel>Mega Man X</IonLabel>
39       </IonItem>
40       <IonItem>
41         <IonLabel>The Legend of Zelda</IonLabel>
42       </IonItem>
43       <IonItem>
44         <IonLabel>Pac-Man</IonLabel>
45       </IonItem>
46       <IonItem>
47         <IonLabel>Super Mario World</IonLabel>
48       </IonItem>
49     </IonList>
50   );
51 }
52
53 export default Example;

```

54 Komponen ini dapat ditambahkan properti ‘inset’ untuk memberikan *margin* pada tiap tepi, 55 ‘lines’ untuk memberikan *border* bawah, dan juga ‘mode’ untuk menentukan *style* platform apa 56 yang diinginkan. Secara *default*, properti ‘inset’ memiliki nilai ‘false’, properti ‘lines’ memiliki nilai 57 ‘undefined’, dan juga properti ‘mode’ memiliki nilai ‘undefined’. Pengembang dapat mengubah nilai 58 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’

1 Menu

- 2 Menu merupakan *navigation drawer* yang meluncur dari sisi tampilan saat dijalankan. Komponen
 3 ini adalah pola navigasi umum yang dapat secara permanen ditampilkan di layar saat diperlukan.
 4 Komponen Menu dapat digunakan untuk membuat tata letak aplikasi yang lebih terstruktur dan
 5 meningkatkan pengalaman pengguna dengan menyediakan akses mudah ke berbagai bagian dari
 6 aplikasi. Cara penggunaan Menu terdapat pada (Kode 2.19).

Kode 2.19: Contoh Kode Penggunaan Menu

```

7 import React from 'react';
8 import { IonButtons, IonContent, IonHeader, IonMenu, IonMenuButton, IonPage, IonTitle, IonToolbar } from '@ionic/react';
9 function Example() {
10   return (
11     <>
12       <IonMenu contentId="main-content">
13         <IonHeader>
14           <IonToolbar>
15             <IonTitle>Menu Content</IonTitle>
16           </IonToolbar>
17         </IonHeader>
18         <IonContent className="ion-padding">This is the menu content.</IonContent>
19       </IonMenu>
20       <IonPage id="main-content">
21         <IonHeader>
22           <IonToolbar>
23             <IonButtons slot="start">
24               <IonMenuButton></IonMenuButton>
25             </IonButtons>
26             <IonTitle>Menu</IonTitle>
27           </IonToolbar>
28         </IonHeader>
29         <IonContent className="ion-padding">Tap the button in the toolbar to open the menu.</IonContent>
30       </IonPage>
31     </>
32   );
33 }
34 export default Example;

```

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

42 Modal

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

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

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

Kode 2.20: Contoh Kode Penggunaan Modal

```

9 import React, { useState } from 'react';
10 import { IonButtons, IonButton, IonModal, IonHeader, IonContent, IonToolbar, IonTitle, IonPage } from '@ionic/react';
11
12
13 function Example() {
14   const [isOpen, setIsOpen] = useState(false);
15
16   return (
17     <IonPage>
18       <IonHeader>
19         <IonToolbar>
20           <IonTitle>Inline Modal</IonTitle>
21         </IonToolbar>
22       </IonHeader>
23       <IonContent className="ion-padding">
24         <IonButton expand="block" onClick={() => setIsOpen(true)}>
25           Open
26         </IonButton>
27         <IonModal isOpen={isOpen}>
28           <IonHeader>
29             <IonToolbar>
30               <IonTitle>Modal</IonTitle>
31               <IonButtons slot="end">
32                 <IonButton onClick={() => setIsOpen(false)}>Close</IonButton>
33               </IonButtons>
34             </IonToolbar>
35           </IonHeader>
36           <IonContent className="ion-padding">
37             <p>
38               Lorem ipsum dolor sit amet consectetur adipisicing elit. Magni illum quidem recusandae ducimus quos
39               reprehenderit. Veniam, molestias quos, dolorum consequuntur nisi deserunt omnis id illo sit cum qui.
40               Eaque, dicta.
41             </p>
42           </IonContent>
43         </IonModal>
44       </IonContent>
45     </IonPage>
46   );
47 }
48
49 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.

- 1 – Signature: onWillDismiss<T = any>() => Promise<OverlayEventDetail<T>>
- 2 • present
 - 3 – Deskripsi: Menampilkan modal setelah dibuat.
 - 4 – Signature: present() => Promise<void>
- 5 • setCurrentBreakpoint
 - 6 – Deskripsi: Memindahkan modal gaya lembar ke breakpoint tertentu. Nilai breakpoint
7 harus sesuai dengan yang didefinisikan dalam array breakpoints.
 - 8 – Signature: setCurrentBreakpoint(breakpoint: number) => Promise<void>

9 Router

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

16 Toolbar

17 Toolbar merupakan komponen yang digunakan untuk menampilkan judul, tombol, ikon, tombol
 18 kembali, tombol menu, kotak pencarian, segmen, dan indikator proses di aplikasi. Toolbars umumnya
 19 ditempatkan di atas atau di bawah konten dan menyediakan konten dan tindakan untuk layar saat
 20 ini. Ketika toolbar ditempatkan di *header*, Toolbar akan muncul di bagian atas konten, sedangkan
 21 jika ditempatkan di *footer*, Toolbar akan muncul di bagian bawah. Cara menggunakan Toolbar
 22 terdapat pada (Kode 2.21).

Kode 2.21: Contoh Kode Penggunaan Toolbar

```

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

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

2.3.2 Capacitor Native

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

Camera

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

Kode 2.22: Kode untuk Menginstal Plugin Camera

```
15 16 17 npm install @capacitor/camera
```

Kode 2.23: Contoh Kode Capacitor Camera

```
18 import { Camera, CameraResultType } from '@capacitor/camera';
19
20
21 const takePicture = async () => {
22   const image = await Camera.getPhoto({
23     quality: 90,
24     allowEditing: true,
25     resultType: CameraResultType.Uri
26   });
27
28   // image.webPath will contain a path that can be set as an image src.
29   // You can access the original file using image.path, which can be
30   // passed to the Filesystem API to read the raw data of the image,
31   // if desired (or pass resultType: CameraResultType.Base64 to getPhoto)
32   var imageUrl = image.webPath;
33
34   // Can be set to the src of an image now
35   imageElement.src = imageUrl;
36
37 };
38
```

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

Filesystem

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

Kode 2.24: Kode untuk Menginstal Plugin Filesystem

```
1 31 npm install @capacitor/action-sheet
```

Kode 2.25: Contoh Kode Penggunaan Filesystem

```
4 51 import { Filesystem, Directory, Encoding } from '@capacitor/filesystem';
6 52
7 53 const writeSecretFile = async () => {
8 54   await Filesystem.writeFile({
9 55     path: 'secrets/text.txt',
10 56     data: 'This is a test',
11 57     directory: Directory.Documents,
12 58     encoding: Encoding.UTF8,
13 59   });
14 60 };
15 61
16 62 const readSecretFile = async () => {
17 63   const contents = await Filesystem.readFile({
18 64     path: 'secrets/text.txt',
19 65     directory: Directory.Documents,
20 66     encoding: Encoding.UTF8,
21 67   });
22 68
23 69   console.log('secrets:', contents);
24 70 };
25 71
26 72 const deleteSecretFile = async () => {
27 73   await Filesystem.deleteFile({
28 74     path: 'secrets/text.txt',
29 75     directory: Directory.Documents,
30 76   });
31 77 };
32 78
33 79 const readFilePath = async () => {
34 80   // Here's an example of reading a file with a full file path. Use this to
35 81   // read binary data (base64 encoded) from plugins that return File URIs, such as
36 82   // the Camera.
37 83   const contents = await Filesystem.readFile({
38 84     path: 'file:///var/mobile/Containers/Data/Application/22A433FD-D82D-4989-8BE6-9FC49DEA20BB/Documents/text.txt',
39 85   });
40 86
41 87   console.log('data:', contents);
42 88 };
```

44 Preference

45 Plugin Preference adalah *plugin* yang berguna untuk menyimpan data sederhana dalam bentuk kunci
 46 atau nilai yang dapat diakses secara bersamaan. Preferences API menyediakan area penyimpanan
 47 data yang mendukung kunci atau nilai untuk aplikasi Ionic. Untuk menggunakan Plugin Preference
 48 dalam aplikasi Ionic, perlu meng-*install plugin* Capacitor Preferences dengan cara yang terdapat
 49 pada (Kode 2.26), sedangkan cara untuk menggunakan Capacitor terdapat pada (Kode 2.27).

Kode 2.26: Kode untuk Menginstal Plugin Preference

```
50 51 npm install @capacitor/preferences
```

Kode 2.27: Contoh Kode Plugin Preference

```
53 54 1 import { Preferences } from '@capacitor/preferences';
55 56
56 57 const setName = async () => {
57 58   await Preferences.set({
58 59     key: 'name',
59 60     value: 'Max',
60 61   });
61 62 };
62 63
63 64 const checkName = async () => {
64 65   const { value } = await Preferences.get({ key: 'name' });
65 66 }
```

```

13|   console.log(`Hello ${value}`);
14| };
15|
16| const removeName = async () => {
17|   await Preferences.remove({ key: 'name' });
18| };

```

8 2.4 RSS [3]

9 RSS merupakan singkatan dari Really Simple Syndication adalah format sindikasi konten web yang
 10 merupakan dialek dari XML. Semua file RSS harus disesuaikan dengan spesifikasi XML 1.0 yang
 11 dipublikasikan oleh World Wide Web Consortium (W3C). Dokumen RSS terdiri dari elemen <rss>
 12 yang memiliki atribut wajib bernama version untuk menentukan versi RSS yang digunakan. Di
 13 bawah elemen <rss> terdapat elemen <channel> yang berisi informasi tentang saluran (metadata)
 14 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.

15 2.5 React-Axios [4]

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

- 18 • Mewarisi semua fitur utama dari Axios.
- 19 • Berbasis komponen React.
- 20 • Menggunakan callback fungsi anak untuk menangani hasil permintaan (error, response, isLoading, makeRequest, axios).
- 21 • Membatalkan permintaan sebelumnya secara otomatis.
- 22 • Debounce untuk mencegah permintaan berulang dengan cepat.
- 23 • Permintaan hanya dipicu saat ada perubahan properti dan dalam keadaan siap (isReady).
- 24 • Mendukung callback untuk sukses (onSuccess), kesalahan (onError), dan loading (onLoading).
- 25 • Mendukung instance Axios kustom melalui properti atau <AxiosProvider>.
- 26 • Cara untuk melakukan install package React-Axios terdapat pada Kode 2.28.

Kode 2.28: Kode untuk Menginstall Komponen React-Axios

```

28 1 npm install react-axios

```

31 Komponen yang dimiliki oleh React-Axios meliputi komponen dasar dan juga komponen pem-
 32 bantu.

- 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)

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

```

91 // Post a request for a user with a given ID
102 render() {
113   return (
124     <div>
135       <Get url="/api/user" params={{id: "12345"}>
146         {(error, response, isLoading, makeRequest, axios) => {
157           if(error) {
168             return (<div>Something bad happened: {error.message} <button onClick={() => makeRequest({ params: { reload: true } })}>Retry</button></div>)
179           }
190           else if(isLoading) {
201             return (<div>Loading...</div>)
212           }
223           else if(response !== null) {
234             return (<div>{response.data.message} <button onClick={() => makeRequest({ params: { refresh: true } })}>Refresh</button></div>)
245           }
266             return (<div>Default message before request is made.</div>)
277           }
288         </Get>
299       </div>
300     )
321   }

```

33 2.6 React Native XML to JSON Conversion [5]

34 Package React Native XML to JSON Conversion merupakan konverter sederhana dari XML ke
 35 objek JavaScript dan mendukung konversi dua arah menggunakan sax-js dan xmlbuilder-js. Package
 36 ini diciptakan dari hasil fork pada node-xml2js yang mendukung React Native sehingga mendukung
 37 platform Android dan juga iOS.

38 Cara untuk melakukan install package React-Native-XML2JS terdapat pada Kode 2.30.

Kode 2.30: Kode untuk Menginstal Komponen React-Axios

```

39 401 npm install react-native-xml2js

```

42 Terdapat juga contoh penggunaan XML2JS terdapat pada Kode 2.31.

Kode 2.31: Kode Contoh Penggunaan XML2JS

```

43 441 var parseString = require('react-native-xml2js').parseString;
452 var xml = "<root>Hello_xml2js!</root>"
463 parseString(xml, function (err, result) {
474   console.dir(result);
485 });

```

1

BAB 3

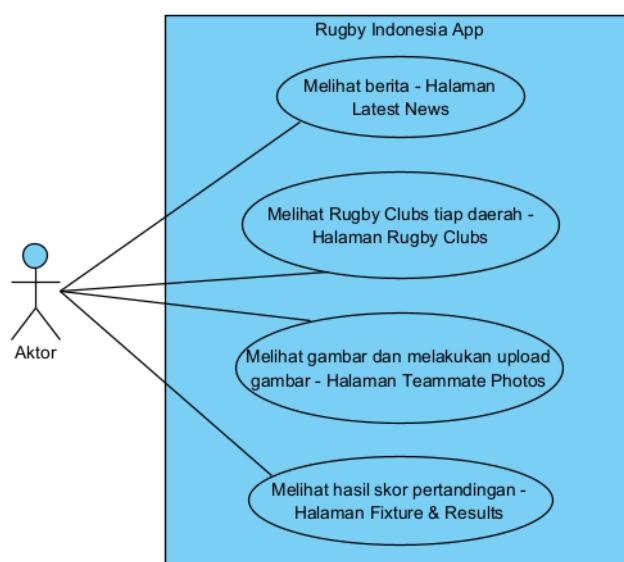
2

ANALISIS

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

6 3.1 Analisis Sistem Kini

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



Gambar 3.1: Use Case Diagram Rugby Indonesia App

1 3.1.1 Halaman Utama

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

Tabel 3.1: Tabel Skenario Awal dari Aplikasi Rugby Indonesia

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

8 3.1.2 Halaman Teammate Photos

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

Tabel 3.2: Tabel Skenario dari Halaman Teammate Photos

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

13 3.1.3 Halaman Rugby Clubs

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

Tabel 3.3: Tabel Skenario dari Halaman Rugby Clubs

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

¹ 3.1.4 Halaman Fixture & Results

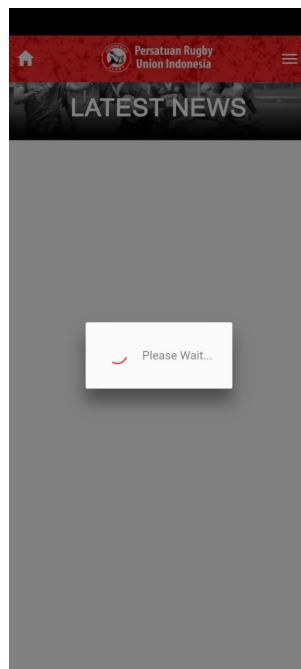
- ² Pada halaman Fixture & Results, pengguna dapat melihat hasil pertandingan skor dari permainan ³ rugby yang sudah diselenggarakan. Skenario pengguna saat berada pada halaman Fixture & Results ⁴ terdapat pada Tabel 3.4.

Tabel 3.4: Tabel Skenario dari Halaman Fixture & Results

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

⁵ 3.2 Analisis Masalah Aplikasi Rugby Indonesia

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



Gambar 3.2: Halaman Utama Aplikasi Rugby Indonesia

3.3 Analisis RSS Rugby Indonesia

Situs web Rugby Indonesia saat ini memiliki RSS bertipe XML yang terdapat pada <https://dnartworks.rugbyindonesia.or.id/indonesianrugby/news/list.xml>. RSS tersebut dapat digunakan untuk melakukan pembangunan aplikasi Rugby Indonesia dengan cara mengambil beberapa objek seperti title dan description yang terdapat pada RSS tersebut. tangkapan layar RSS dari Rugby Indonesia terdapat pada Kode 3.1.

Kode 3.1: Potongan Kode RSS Rugby Indonesia

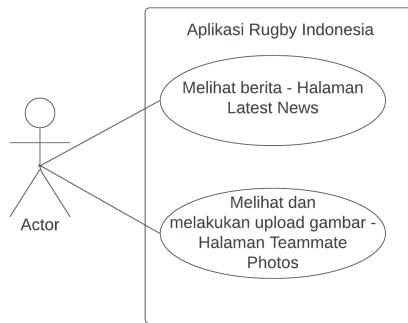
```

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        <pubDate>Tue, 30 Jan 2024 06:46:29 +0000</pubDate>
34        <category>
35        <![CDATA[ Rugby Masuk sekolah ]]>
36      </dc:creator>
37      <category>
38        <![CDATA[ hambali ]]>
39      </category>
40      <guid isPermaLink="false">https://rugbyindonesia.or.id/?p=6435</guid>
41      <description>
42        <![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 --> ]]>
43      </description>
44      <content:encoded>
45        <![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 --> ]]>
46    </content:encoded>
47  </item>
48  </channel>
49</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 dipangkas sehingga tidak memiliki halaman-halaman tersebut. Maka dari itu, fungsi dari aplikasi Rugby Indonesia akan berubah, dan

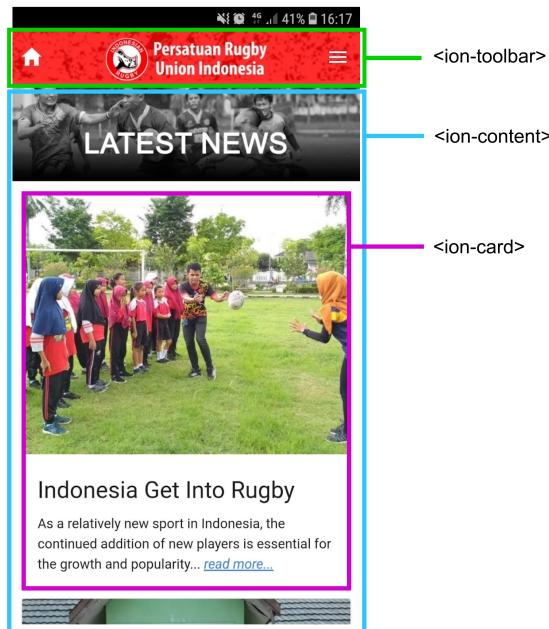
- 1 dapat dilihat seperti pada Gambar 3.3.



Gambar 3.3: UCD 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.

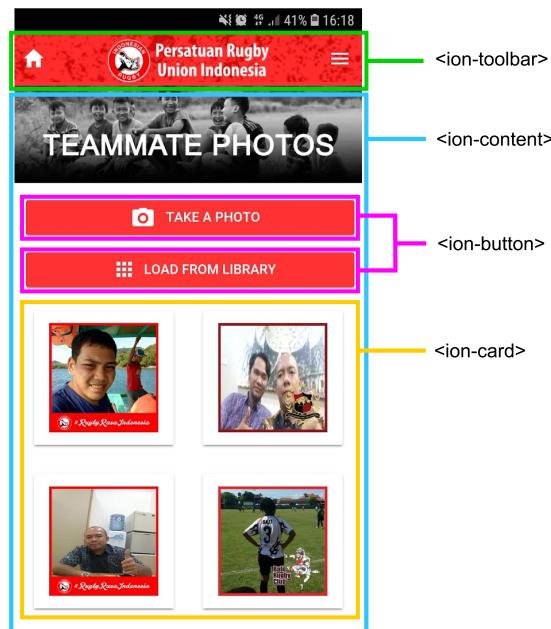
Tabel 3.5: Tabel Skenario saat Pengguna Mengupload Gambar

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

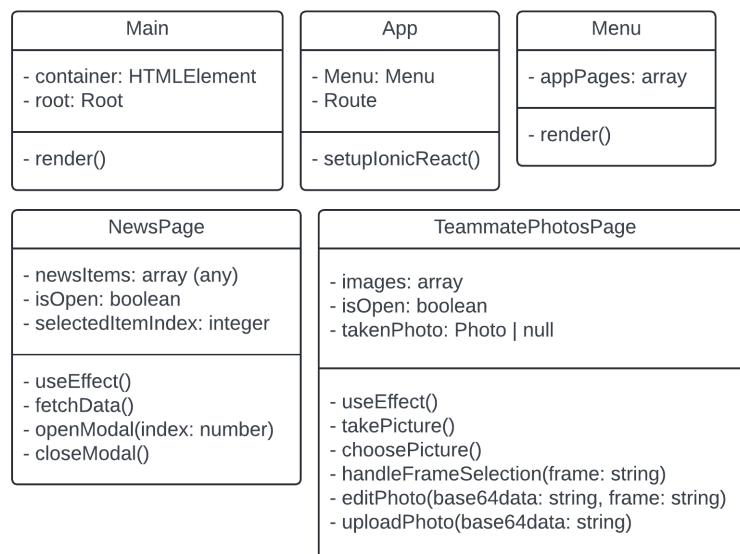
BAB 4

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
15 React.
16 • Method render: melakukan render pada aplikasi React.

17 2. Kelas App

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

- 5 • Atribut menu: memanggil kelas menu.
- 6 • Atribut route: melakukan routing terhadap path yang ingin dibuka.
- 7 • Method setupIonicReact: melakukan setup pada aplikasi dengan menggunakan IonicCon-
8 fig yang sudah disediakan dari library.

9 3. Kelas Menu

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

- 14 • Atribut appPages: berfungsi untuk menyimpan halaman dari aplikasi yang disimpan ke
15 dalam array.
- 16 • Method render: berfungsi untuk melakukan render menu pada aplikasi Rugby Indonesia.

17 4. Kelas NewsPage

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

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

32 5. Kelas TeammatePhotosPage

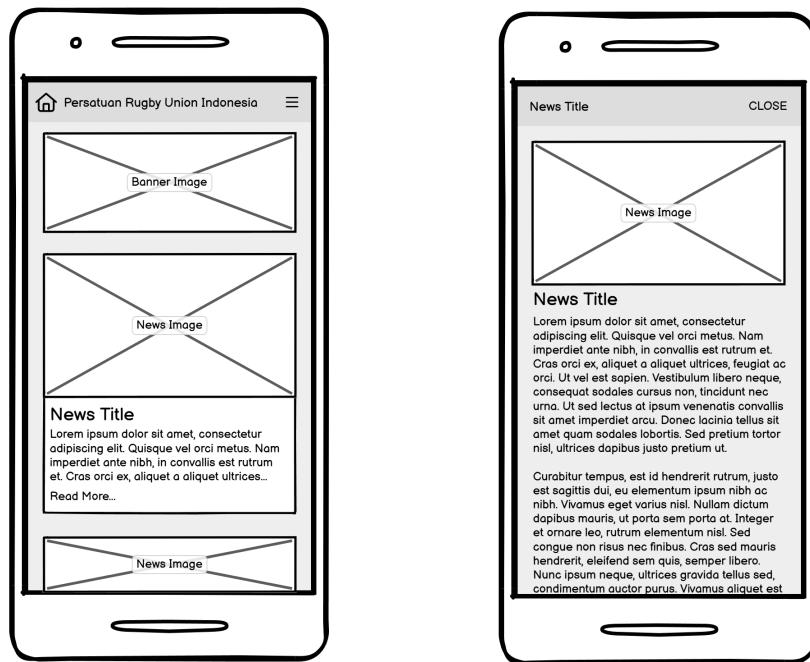
33 Kelas TeammatePhotosPage bertugas untuk menampilkan foto-foto yang telah diunggah
34 oleh pengguna aplikasi Rugby Indonesia, serta melakukan *upload* foto agar foto tersebut
35 dapat dilihat oleh pengguna aplikasi Rugby Indonesia lainnya dengan cara mengambil foto
36 tersebut secara langsung atau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

- 40 • State images: berfungsi untuk menyimpan foto yang diambil menggunakan protokol RSS.
- 41 • State isOpen: berfungsi untuk menentukan apakah modal dalam keadaan terbuka atau
42 tidak.

- 1 • takenPhoto: berfungsi untuk menyimpan dahulu foto sebelum pengguna melakukan pemilihan frame.
- 2 • Method useEffect: melakukan pengambilan data yang berasal dari sistem eksternal.
- 3 • Method takePicture: melakukan pengambilan gambar menggunakan kamera.
- 4 • Method choosePicture: melakukan pengambilan gambar yang berasal dari galeri.
- 5 • Method handleFrameSelection: melakukan penanganan terhadap frame yang sudah dipilih untuk dikirim ke method editPhoto.
- 6 • Method editPhoto: melakukan penggabungan antara foto dan juga frame yang telah dipilih oleh pengguna sebelum gambar tersebut iupload.
- 7 • Method uploadPhoto: melakukan pengunggahan gambar ke server menggunakan metode post.

12 4.2 Perancangan Antarmuka

- 13 Berikut ini merupakan perancangan antarmuka dari aplikasi Rugby Indonesia untuk pengguna dari
14 aplikasi ini.



(a) Perancangan Antarmuka Halaman Latest News

(b) Perancangan Antarmuka Halaman Deskripsi Berita

Gambar 4.2: Perancangan Antarmuka Bagian Latest News

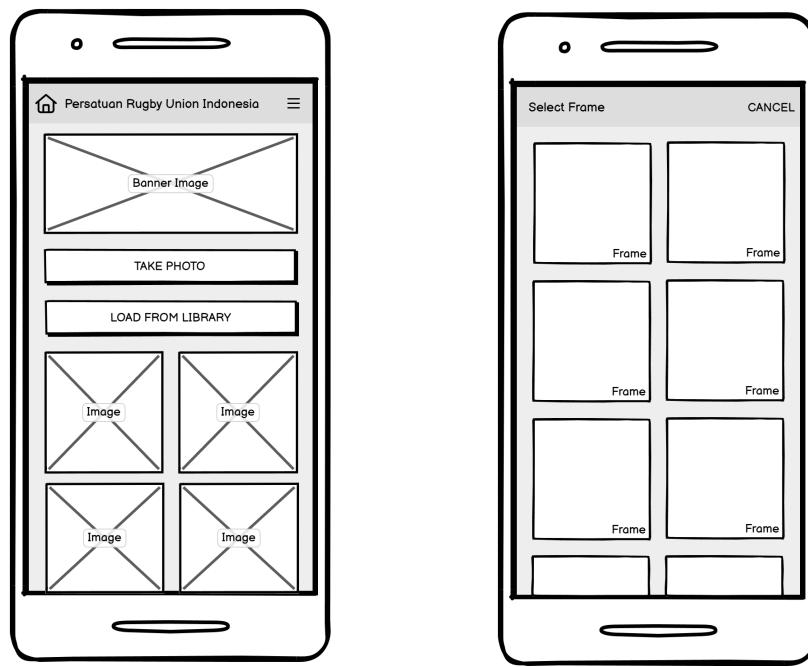
15 4.2.1 Perancangan Antarmuka Halaman Latest News

- 16 Pada halaman Latest News, informasi yang akan ditampilkan oleh halaman ini berisi berita-berita
17 terkini seputar Rugby Indonesia. Gambar perancangan dari halaman Latest News dapat dilihat
18 pada Gambar 4.2a. Pada gambar tersebut, terdapat toolbar dari halaman Latest News yang berisi
19 tombol home, nama dari aplikasi, dan juga tombol menu. Di bawah toolbar, terdapat banner dari
20 halaman Latest News. Lalu terdapat konten yang berisi dari gambar headline berita, judul berita,

- 1 dan deskripsi singkat dari berita tersebut. Pengguna dapat melihat deskripsi lengkap dari berita
 2 tersebut dengan melakukan klik pada tulisan “Read More...” yang terletak di bawah deskripsi
 3 singkat berita.

4 4.2.2 Perancangan Antarmuka Deskripsi Berita

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



Gambar 4.3: Perancangan Antarmuka Bagian Teammate Photos

10 4.2.3 Perancangan Antarmuka Halaman Teammate Photos

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

15 4.2.4 Perancangan Antarmuka Halaman Frame

- 16 Halaman ini hanya akan menampilkan pilihan frame yang dapat digunakan oleh pengguna Rugby
 17 Indonesia setelah pengguna melakukan pengambilan gambar menggunakan kamera atau melakukan
 18 pemilihan gambar yang berasal dari galeri. Gambar dari halaman ini dapat dilihat pada Gambar

- 1 4.3b, di mana pada gambar tersebut terdapat tombol “CANCEL” apabila pengguna tidak jadi
- 2 melakukan pengunggahan gambar dan juga pilihan frame yang tersedia.

3 4.3 Perancangan Struktur HTML

- 4 Struktur HTML dari setiap halaman memiliki struktur yang sama, di mana pada setiap halaman
- 5 terdapat header dan juga content. Header dan content yang terdapat pada struktur ini memiliki
- 6 penjelasan sebagai berikut:

7 4.3.1 Header

- 8 Header dari setiap halaman memiliki struktur yang sama, di mana pada setiap halaman, header
- 9 menggunakan komponen <IonHeader> dan memiliki komponen <IonToolbar>, <IonButtons>,
- 10 dan juga <IonTitle> di dalam header tersebut. Hal yang membedakan dari tiap header hanya
- 11 berupa judul yang berada di dalam komponen <IonTitle> dan juga button yang terdapat pada
- 12 header tersebut.

13 4.3.2 Content

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

- 17 • Halaman Latest News

18 Content yang terdapat pada halaman Latest News berisi banner dari halaman Latest News, dan
19 juga berita terkini seputar Rugby Indonesia. Pada tiap berita yang berada di halaman Latest
20 News, berita tersebut dibungkus di dalam komponen <IonCard> yang memiliki komponen
21 <IonCardHeader> sebagai header dari berita dan juga komponen <IonCardContent> sebagai
22 konten dari berita tersebut.

- 23 • Halaman Teammate Photos

24 Content yang terdapat pada halaman Teammate Photos berisi banner dari halaman Teammate
25 Photos, tombol untuk melakukan pengambilan gambar menggunakan kamera, tombol untuk
26 melakukan pengunggahan gambar dari galeri, serta komponen <PhotoGallery> yang berisi
27 gambar yang telah diunggah oleh pengguna.

¹

BAB 5

²

IMPLEMENTASI DAN PENGUJIAN

³ 5.1 Spesifikasi Perangkat Lunak

- 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 News Page

- ¹⁵ Ketika pengguna telah mengunduh aplikasi Rugby Indonesia dan membukanya, aplikasi tersebut
¹⁶ akan menampilkan halaman Latest News seperti yang terdapat pada Gambar [5.1](#).



Gambar 5.1: Halaman Latest News pada Aplikasi Rugby Indonesia

1 5.2.2 Implementasi Antarmuka News Detail

- 2 Halaman ini muncul ketika pengguna telah melakukan klik pada tulisan “Read More...” yang tertera
- 3 di bawah deskripsi singkat dari masing-masing berita. Tampilan dari halaman ini terdapat pada
- 4 Gambar 5.2.



Gambar 5.2: Halaman Deskripsi Berita pada Aplikasi Rugby Indonesia

5 5.2.3 Implementasi Antarmuka Teammate Photos Page

- 6 Halaman ini ditampilkan ketika pengguna telah melakukan klik pada tombol menu di kanan atas
- 7 pada halaman Latest News dan memilih menu Teammate Photos. Tampilan dari halaman ini

- 1 terdapat pada Gambar 5.3.



Gambar 5.3: Halaman Teammate Photos pada Aplikasi Rugby Indonesia

2 5.2.4 Implementasi Antarmuka Halaman Frame

- 3 Halaman ini ditampilkan ketika pengguna telah melakukan pengambilan foto dengan melakukan
4 klik pada tombol “TAKE PHOTO” yang terdapat pada halaman Teammate Photos atau telah
5 melakukan pemilihan foto dari galeri dengan melakukan klik pada tombol “UPLOAD FROM
6 LIBRARY” yang terdapat pada halaman Teammate Photos juga. Tampilan dari halaman ini
7 terdapat pada Gambar 5.4



Gambar 5.4: Halaman Pemilihan Frame pada Aplikasi Rugby Indonesia

1 5.3 Pengujian Fungsional

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

4 5.3.1 Pengujian Melihat Berita

Tabel 5.1: Tabel Pengujian Melihat Berita

No.	Langkah pengujian	Hasil yang diharapkan	Status
1	Menekan tombol home pada toolbar	Kembali ke awal halaman Latest News	Berhasil
2	Menekan tombol menu pada toolbar	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 toolbar di halaman News Detail	Kembali ke halaman Latest News	Berhasil

5 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 home pada toolbar	Kembali ke halaman Latest News	Berhasil
2	Menekan tombol menu pada toolbar	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 frame	Berhasil
5	Memilih salah satu frame	Kembali ke halaman Teammate Photos dan menampilkan gambar yang telah diunggah	Berhasil
6	Menekan tombol “CANCEL” pada halaman pemilihan frame	Kembali ke halaman Teammate Photos 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 frame	Berhasil

6 5.4 Pengujian oleh Pengguna Umum

- 7 Hasil pengujian berikut ini dilakukan oleh pengguna umum (*User Acceptance Test*) di mana
 8 pengujian ini dilakukan oleh 6 orang responden. Responden yang melakukan pengujian kali ini
 9 merupakan penggemar olahraga dengan rentang umur 21 hingga 23 tahun yang memiliki *smartphone*
 10 dengan sistem operasi Android. Responden melakukan pengujian aplikasi Rugby Indonesia dengan

- 1 cara responden mengunduh aplikasi Rugby Indonesia dari tautan yang terdapat pada Google Form
 2 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 Latest News saat pertama kali dibuka?	6 menjawab Ya
2	Apakah seluruh berita yang terdapat pada halaman Latest News ditampilkan dengan baik?	5 menjawab Ya, 1 menjawab Tidak
3	Apakah tombol "TAKE PHOTO" dapat membuka kamera smartphone?	6 menjawab Ya
4	Apakah tombol "LOAD FROM LIBRARY" dapat membuka galeri smartphone?	6 menjawab Ya
5	Apakah Anda mengalami kendala saat melakukan pengambilan gambar atau mengupload gambar dari galeri menggunakan aplikasi Rugby Indonesia?	6 menjawab Tidak
6	Apakah Anda mengalami <i>crash, forced close</i> , atau kendala lain saat menggunakan aplikasi Rugby Indonesia terbaru?	6 menjawab Tidak

- 3 Dari hasil pada Tabel 5.3, terdapat beberapa saran yang diberikan oleh responden, yaitu:
 4 • Tata cara penulisan jangan terlalu panjang.
 5 • Terdapat *caption* pada gambar berita.
 6 • Foto yang terdapat pada halaman Teammate Photos dapat diperbesar.
 7 • 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 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].

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               <IonMenuItemToggle 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               </IonMenuItemToggle>
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 fetchData();
51 [], []);
52
53 const openModal = (index: number) => {
54     setSelectedItemIndex(index);
55     setIsOpen(true);
56 };
57
58 const closeModal = () => {
59     setIsOpen(false);
60     setSelectedItemIndex(-1);
61 };
62
63 return (
64     <IonPage>
65         <IonHeader>
66             <IonToolbar>
67                 <IonButtons slot="start">
68                     <a href="/latest_news">
69                         <img alt="home-icon" className='home-icon' />
70                     </a>
71                 </IonButtons>
72                 <IonButtons slot="end">
73                     <IonMenuButton />
74                 </IonButtons>
75             <IonTitle>
76                 Persatuan Rugby
77                 Union Indonesia
78             </IonTitle>
79         </IonToolbar>
80     </IonHeader>
81     <IonContent className="ion-padding">
82         <div className="news-section-image">
83             <img alt='latest-news-banner' />
84         </div>
85
86         {newsItems.map((item, index) => (
87             <IonCard key={index}>
88                 <img alt='latest-news-image' src={item.thumbnail} />
89                 <IonCardHeader>
90                     <IonCardTitle>{item.title}</IonCardTitle>
91                 </IonCardHeader>
92                 <IonCardContent>
93                     <div dangerouslySetInnerHTML={{ __html: item.description }}></div>
94                     <a onClick={() => openModal(index)}>Read More...</a>
95                 </IonCardContent>
96             </IonCard>
97         )));
98     <IonModal isOpen={isOpen} onDidDismiss={closeModal}>
99         <IonHeader>
100             <IonToolbar>
101                 <IonTitle>{selectedItemIndex !== -1 ? newsItems[selectedItemIndex].title : ''}</IonTitle>
102                 <IonButtons slot="end">
103                     <IonButton onClick={closeModal}>CLOSE</IonButton>
104                 </IonButtons>
105             </IonToolbar>
106         </IonHeader>
107         <IonContent className="ion-padding">
108             <img alt='latest-news-image' src={selectedItemIndex !== -1 ? newsItems[selectedItemIndex].thumbnail : ''}>
109             <h1>{selectedItemIndex !== -1 ? newsItems[selectedItemIndex].title : ''}</h1>
110             {selectedItemIndex !== -1 && (
111                 <div dangerouslySetInnerHTML={{ __html: newsItems[selectedItemIndex].content }}></div>
112             )}
113         </IonContent>
114     </IonModal>
115     </IonContent>
116   </IonPage>
117 );
118
119 export default NewsPage;
120
121

```

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,${base64data}';
101  );
102};
103
104 const uploadPhoto = async (base64data: string) => {
105   try {
106     const userId = "anonymous";
107
108     const response = await fetch('https://dnartworks.rugbyindonesia.or.id/indonesianrugby/photos/upload.json', {
109       method: 'POST',
110       headers: {
111         'Content-Type': 'application/x-www-form-urlencoded'
112       },
113       body: `userId=${userId}&photo=${base64data}`
114     });
115
116     if (response.ok) {
117       console.log('Photo uploaded successfully!');
118     } else {
119       console.error(`Failed to upload photo: ${response.statusText}`);
120     }
121   } catch (error) {
122     console.error(`Error uploading photo: ${error}`);
123   }
124 };
125
126 const handleFrameSelection = async (frame: string) => {
127   if (takenPhoto) {
128     const editedBase64Data = await editPhoto(takenPhoto.base64String!, frame);
129     await uploadPhoto(editedBase64Data);
130     window.location.reload();
131   }
132   setIsOpen(false);
133 };
134
135 return (
136   <IonPage>
137     <IonHeader>
138       <IonToolbar>
139         <IonButtons slot="start">
140           <a href="/latest_news">
141             <img src={homeIcon} alt="home-icon" className='home-icon' />
142           </a>
143         </IonButtons>
144         <IonButtons slot="end">
145           <IonMenuButton />
146         </IonButtons>
147         <IonTitle>
148           Persatuan Rugby
149           Union Indonesia
150         </IonTitle>
151       </IonToolbar>
152     </IonHeader>
153     <IonContent className="ion-padding">
154       <div className='teammate-photos-banner'>
155         <img src={bannerImage} alt="Teammate-Photos-Banner" />
156       </div>
157       <br/><br/>
158       <IonButton color="primary" expand="block" onClick={takePicture}>
159         <IonIcon slot="start" icon={camera}></IonIcon>
160         Take Photo
161       </IonButton>
162
163       <IonButton color='primary' expand='block' onClick={choosePicture}>
164         <IonIcon slot="start" icon={appsSharp}></IonIcon>
165         Load from Library
166       </IonButton>
167
168       <PhotoGallery photos={images} />
169
170       <IonModal isOpen={isOpen} onDidDismiss={() => setIsOpen(false)}>
171         <IonHeader>
172           <IonToolbar>
173             <IonTitle>Select Frame</IonTitle>
174             <IonButtons slot="end">
175               <IonButton onClick={() => setIsOpen(false)}>CANCEL</IonButton>
176             </IonButtons>
177           </IonToolbar>
178         </IonHeader>
179         <IonContent className="ion-padding">
180           <div className="frame-selection">
181             {frames.map((frame, index) => (
182               <IonCol size = "6" key={index}>
183                 <img
184                   key={index}
185                   src={frame}
186                   alt={'Frame ${index + 1}'}
187                   className="frame-option"
188                 </img>
189             ))}
190           </IonContent>
191         </IonModal>
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 }

```