

# SKRIPSI

## PEMBUATAN ULANG APLIKASI RUGBY INDONESIA DENGAN IONIC 7 DAN CAPACITOR



Nathanael Adi Trianto

NPM: 6181901041

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



# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>iii</b>
<b>DAFTAR GAMBAR</b>	<b>v</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	3
1.3 Tujuan . . . . .	3
1.4 Batasan Masalah . . . . .	3
1.5 Metodologi . . . . .	3
1.6 Sistematika Pembahasan . . . . .	4
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Rugby Indonesia App . . . . .	5
2.2 ReactJS . . . . .	6
2.3 Ionic 7 Framework . . . . .	7
2.3.1 UI Components . . . . .	7
2.3.2 Capacitor Native . . . . .	12
<b>DAFTAR REFERENSI</b>	<b>17</b>
<b>A KODE PROGRAM</b>	<b>19</b>
<b>B HASIL EKSPERIMEN</b>	<b>21</b>



## DAFTAR GAMBAR

1.1	Halaman aplikasi Rugby Indonesia . . . . .	2
2.1	Halaman aplikasi Rugby Indonesia . . . . .	5
B.1	Hasil 1 . . . . .	21
B.2	Hasil 2 . . . . .	21
B.3	Hasil 3 . . . . .	21
B.4	Hasil 4 . . . . .	21



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

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

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

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

---

<sup>1</sup>[https://id.wikipedia.org/wiki/Sepak\\_bola\\_rugbi](https://id.wikipedia.org/wiki/Sepak_bola_rugbi)

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

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

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

<sup>5</sup>[https://en.wikipedia.org/wiki/Indonesia\\_national\\_rugby\\_union\\_team](https://en.wikipedia.org/wiki/Indonesia_national_rugby_union_team)

<sup>6</sup>[https://en.wikipedia.org/wiki/Rugby\\_union\\_in\\_Indonesia](https://en.wikipedia.org/wiki/Rugby_union_in_Indonesia)

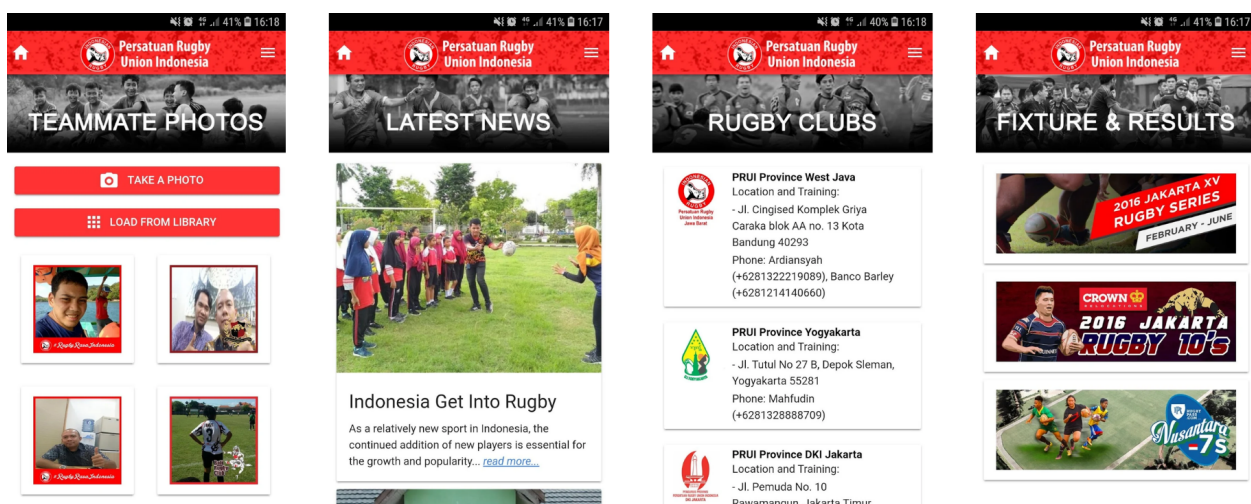
<sup>7</sup><https://rugbyindonesia.or.id/>

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

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

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

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



Gambar 1.1: Halaman-halaman dari aplikasi Rugby Indonesia

Pada saat ini, aplikasi tersebut masih tersedia di Google Play Store<sup>9</sup>, namun aplikasi tersebut tidak dapat dipasang pada perangkat android saat ini dikarenakan *website* <https://rugbyindonesia.>

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

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



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

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

- Halaman *Latest News*, di mana pengguna dapat melihat berita terbaru seputar Rugby Indonesia.
- Halaman *Teammate Photos* di mana pengguna dapat mengambil foto dan langsung mengunggahnya ke dalam halaman *teammate photos*.

## 1.2 Rumusan Masalah

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

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

## 1.3 Tujuan

Tujuan yang ingin dicapai pada penulisan tugas akhir ini yaitu:

1. Dapat mengetahui bagaimana Ionic 7 memungkinkan pengembangan aplikasi Rugby Indonesia.
2. Mengidentifikasi cara kerja dari Capacitor pada pembangunan perangkat lunak Rugby Indonesia.

## 1.4 Batasan Masalah

Batasan masalah yang terdapat pada pengerjaan tugas akhir ini yaitu:

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 lunar android dikarenakan peneliti tidak memiliki perangkat lunak yang menggunakan sistem operasi iOS.
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 aplikasi sebelumnya, pengguna hanya dapat melakukan hal tersebut.

## 1.5 Metodologi

Langkah-langkah yang dilakukan dalam pengerjaan tugas akhir ini yaitu:

1. Melakukan studi literatur serta mendalami ReactJS sebagai salah satu perpustakaan JavaScript.
2. Melakukan studi literatur mengenai *framework* Ionic 7 dan juga Capacitor yang terdapat pada Ionic.

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

## 1.6 Sistematika Pembahasan

Penulisan setiap bab pada dokumen tugas akhir ini yaitu:

1. Bab Pendahuluan

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

2. Bab Dasar Teori

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

3. Bab Analisis

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

4. Bab Perancangan

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

5. Bab Implementasi dan Pengujian

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

6. Bab Kesimpulan dan Saran

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

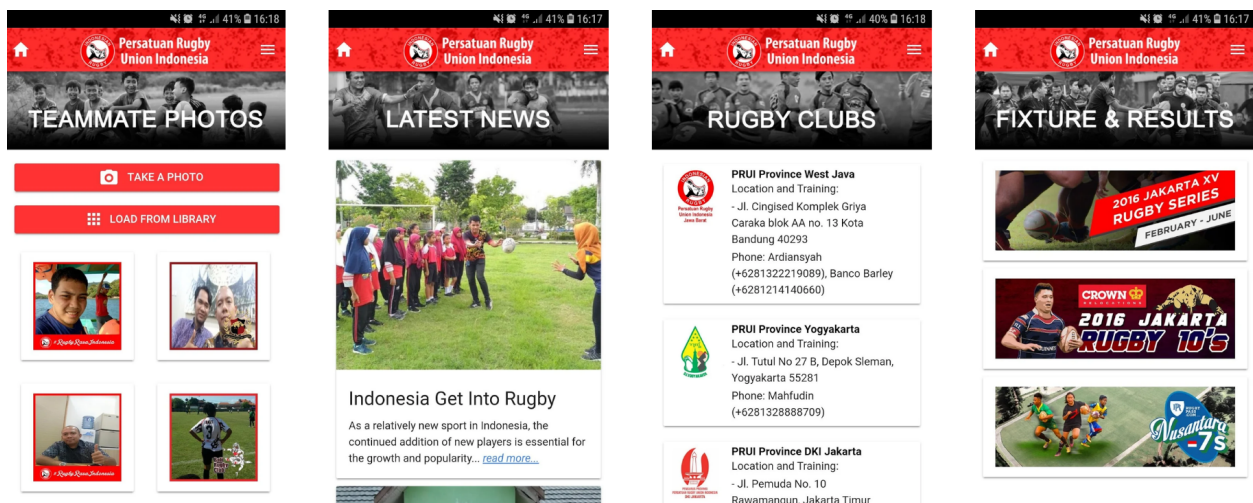
## BAB 2

### LANDASAN TEORI

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

#### 2.1 Rugby Indonesia App

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



Gambar 2.1: Halaman-halaman dari aplikasi Rugby Indonesia

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

1. Halaman *Latest News* yang diambil dari <https://rugbyindonesia.or.id/berita/> dengan memanfaatkan protokol RSS.
2. Halaman *Fixture & Results*.
3. Halaman *Teammate Photos* dengan fungsi:
  - Pengguna dapat langsung mengambil foto dari aplikasi tersebut.

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

## 2.2 ReactJS

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

React dibuat oleh Jordan Walke, seorang insinyur perangkat lunak di Facebook (sekarang bernama Meta), yang merilis prototipe awal React yang disebut "FaxJS". Dia terinspirasi dari bahasa pemrograman XHP, perpustakaan komponen HTML untuk PHP. React pertama kali diterapkan di Facebook's News Feed pada tahun 2011 dan kemudian di Instagram pada tahun 2012. React bersifat open-source di JSConf US pada Mei 2013.

Beberapa fitur dan kelebihan ReactJS antara lain:

- Reusable Components: Dengan ReactJS, Anda bisa menggunakan lagi komponen yang sudah dikembangkan menjadi aplikasi. Sebab, ReactJS adalah library yang open-source, sehingga Anda bisa membangun komponen siap pakai, yang akan mempercepat proses development aplikasi web kompleks.
- Virtual DOM: ReactJS menggunakan Virtual DOM, yang memungkinkan perubahan pada tampilan aplikasi hanya terjadi pada bagian yang berubah saja, tanpa harus merender ulang seluruh tampilan aplikasi. Hal ini membuat aplikasi menjadi lebih cepat dan efisien.
- SEO-Friendly: ReactJS bisa memaksimalkan optimisasi mesin pencari (SEO) aplikasi web dengan meningkatkan performanya. Sebab, implementasi Virtual DOM merupakan salah satu faktor yang memengaruhi kecepatan.
- Learn Once, Write Anywhere: ReactJS tidak membuat asumsi tentang sisa stack pengguna, sehingga pengguna dapat mengembangkan fitur baru di React tanpa menulis ulang kode yang ada. React juga dapat me-render di server dengan menggunakan Node dan menjalankan aplikasi seluler menggunakan React Native.
- UI Interaktif: ReactJS dapat disebut sebagai "Learn One – Write Anywhere" library, karena baik dalam pengembangan aplikasi web dan mobile, React mengikuti pola desain yang sama, memfasilitasi proses transisi. Menggunakan JavaScript polos dan React, Anda dapat membuat UI yang kaya untuk aplikasi asli, serta didukung oleh platform iOS dan Android.

Dengan kelebihan-kelebihan tersebut, ReactJS menjadi pilihan yang masuk akal baik untuk startup maupun perusahaan.

## 2.3 Ionic 7 Framework

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

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

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

### 2.3.1 UI Components

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

#### Action Sheet

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

Kode 2.1: Contoh kode untuk membuat Action Sheet

```
import React from 'react';
import { IonActionSheet, IonButton } from '@ionic/react';

function Example() {
  return (
    <>
      <IonButton id="open-action-sheet">Open</IonButton>
      <IonActionSheet
        trigger="open-action-sheet"
        header="Actions"
        buttons={[
          {
            text: 'Delete',
            role: 'destructive',
            data: {
              action: 'delete',
            },
          },
          {
            text: 'Share',
            data: {
              action: 'share',
            },
          },
        ]}
      />
    </>
  );
}
```

```

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

```

## 17 Accordion

18 Accordion berfungsi untuk mengurangi ruang vertikal dalam mengorganisir informasi yang ingin  
 19 ditampilkan. Accordion memiliki 2 komponen yaitu ion-accordion dan ion-accordion-group. Ke-  
 20 tika menggunakan accordion, ion-accordion harus berada di dalam ion-accordion-group. Contoh  
 21 penggunaan dari Accordion adalah sebagai berikut:

Kode 2.2: Contoh kode untuk membuat Accordion

```

22 import React from 'react';
23 import { IonAccordion, IonAccordionGroup, IonItem, IonLabel } from '@ionic/react';
24 function Example() {
25   return (
26     <IonAccordionGroup>
27       <IonAccordion value="first">
28         <IonItem slot="header" color="light">
29           <IonLabel>First Accordion</IonLabel>
30         </IonItem>
31         <div className="ion-padding" slot="content">
32           First Content
33         </div>
34       </IonAccordion>
35       <IonAccordion value="second">
36         <IonItem slot="header" color="light">
37           <IonLabel>Second Accordion</IonLabel>
38         </IonItem>
39         <div className="ion-padding" slot="content">
40           Second Content
41         </div>
42       </IonAccordion>
43       <IonAccordion value="third">
44         <IonItem slot="header" color="light">
45           <IonLabel>Third Accordion</IonLabel>
46         </IonItem>
47         <div className="ion-padding" slot="content">
48           Third Content
49         </div>
50       </IonAccordion>
51     </IonAccordionGroup>
52   );
53 }
54 export default Example;

```

## 57 Alert

58 Alert pada Ionic berfungsi untuk memberikan informasi serta mengumpulkan informasi dari pengguna  
 59 menggunakan input dari pengguna. Sama seperti Action Sheet, Alert juga biasanya dimunculkan di  
 60 atas konten dari aplikasi, namun Alert biasanya berada di tengah konten aplikasi, sedangkan Action  
 61 Sheet muncul berada dari bawah aplikasi. Contoh dari penggunaan Alert adalah sebagai berikut:

Kode 2.3: Contoh kode untuk membuat Alert

```

11 import React from 'react';
12 import { IonAlert, IonButton } from '@ionic/react';
13
14 function Example() {
15   return (
16     <>
17       <IonButton id="present-alert">Click Me</IonButton>
18       <IonAlert
19         trigger="present-alert"
20         header="A_Short_Title_Is_Best"
21         subHeader="A_Sub_Header_Is_Optional"
22         message="A_message_should_be_a_short,_complete_sentence."
23         buttons={['Action']}
24       ></IonAlert>
25     </>
26   );
27 }
28 export default Example;

```

## 20 Badge

21 Badge adalah elemen inline yang umumnya muncul di dekat dengan elemen lain. Badge ini  
 22 umumnya mengandung angka atau karakter lainnya dan dapat digunakan sebagai elemen widget  
 23 yang menampilkan informasi tambahan tentang elemen induk. Contoh dari penggunaan Badge  
 24 adalah sebagai berikut:

Kode 2.4: Contoh kode untuk membuat Badge

```

25 <IonBadge>11</IonBadge>
26

```

## 28 Breadcrumb

29 Breadcrumb pada Ionic 7 adalah sebuah elemen navigasi tunggal yang merupakan anak dari  
 30 komponen Breadcrumbs. Breadcrumb dapat berupa tautan ke tempat lain dalam aplikasi atau  
 31 berupa teks biasa. Setiap breadcrumb memiliki pemisah di antara mereka dan dapat opsionalnya  
 32 berisi ikon. Breadcrumb digunakan untuk menunjukkan posisi pengguna dalam aplikasi atau situs  
 33 yang besar dan memiliki halaman yang tersusun secara hierarkis. Mereka dapat diklik untuk  
 34 menampilkan popover dengan informasi lebih lanjut atau memperluas breadcrumb yang terlipat.  
 35 Breadcrumb dapat dikonfigurasi dengan berbagai cara, seperti menambahkan ikon, mengatur  
 36 jumlah maksimum item yang ditampilkan, dan mengontrol tampilan item sebelum dan setelah  
 37 penggulungan. Contoh dari penggunaan Breadcrumb adalah sebagai berikut:

Kode 2.5: Contoh kode untuk membuat Breadcrumb

```

38
39 1 import React from 'react';
40 2 import { IonBreadcrumb, IonBreadcrumbs } from '@ionic/react';
41 3 function Example() {
42 4   return (
43 5     <IonBreadcrumbs>
44 6       <IonBreadcrumb href="#home">Home</IonBreadcrumb>
45 7       <IonBreadcrumb href="#electronics">Electronics</IonBreadcrumb>
46 8       <IonBreadcrumb href="#cameras">Cameras</IonBreadcrumb>
47 9       <IonBreadcrumb href="#film">Film</IonBreadcrumb>
48 0     </IonBreadcrumbs>
49 1   );
50 2 }
51 3 export default Example;
52

```

## 1 Button

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

Kode 2.6: Contoh kode untuk membuat Button

```
4 1 <IonButton>Default</IonButton>
```

## 7 Card

8 Card merupakan komponen UI yang digunakan untuk menampilkan konten seperti teks, gam-  
9 bar, tombol, dan daftar dalam sebuah kotak. Komponen ini biasanya terdiri dari header, judul,  
10 gambar, dan konten utama. Card dapat digunakan sebagai komponen tunggal atau digabungkan  
11 dengan komponen lain untuk membuat tampilan yang lebih kompleks. Card dapat disesuaikan  
12 dengan menggunakan properti CSS seperti background dan color. Berikut merupakan contoh dari  
13 penggunaan Card:

Kode 2.7: Contoh kode untuk membuat Card

```
14 15 1 import React from 'react';  
16 2 import { IonCard, IonCardContent, IonCardHeader, IonCardSubtitle, IonCardTitle } from '@ionic/react';  
17 3  
18 4 function Example() {  
19 5   return (  
20 6     <IonCard>  
21 7       <IonCardHeader>  
22 8         <IonCardTitle>Card Title</IonCardTitle>  
23 9         <IonCardSubtitle>Card Subtitle</IonCardSubtitle>  
24 0       </IonCardHeader>  
25 1  
26 2       <IonCardContent>Here's a small text description for the card content. Nothing more, nothing less.</IonCardContent>  
27 3     </IonCard>  
28 4   );  
29 5 }  
30 6 export default Example;
```

## 32 Checkbox

33 Checkbox merupakan komponen yang memungkinkan pengguna untuk memilih beberapa opsi dari  
34 satu set dan muncul sebagai dicentang ketika diaktifkan. Komponen ini digunakan dengan tag  
35 <ion-checkbox>. Contoh dari penggunaan Checkbox adalah sebagai berikut:

Kode 2.8: Contoh kode untuk membuat Checkbox

```
36 1 <IonCheckbox>I agree to the terms and conditions</IonCheckbox>
```

## 39 Chip

40 Chip pada Ionic 7 adalah elemen yang digunakan untuk menampilkan informasi dalam bentuk  
41 container kecil, seperti bubuk. Chip ini dapat berisi berbagai elemen seperti avatar, teks, dan ikon.  
42 Contoh dari penggunaan Chip adalah sebagai berikut:

Kode 2.9: Contoh kode untuk membuat Button

```
43 44 1 import React from 'react';  
45 2 import { IonChip } from '@ionic/react';  
46 3 function Example() {  
47 4   return (  
48 5     <IonChip>
```



```

15   <>
16   <IonChip>Default</IonChip>
17   <IonChip disabled={true}>Disabled</IonChip>
18   <IonChip outline={true}>Outline</IonChip>
19   </>
20   );
21 }
22 export default Example;

```

## 10 Content

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

Kode 2.10: Contoh kode untuk membuat Content

```

15 import React from 'react';
16 import { IonContent } from '@ionic/react';
17
18 function Example() {
19   return (
20     <IonContent className="ion-padding">
21       <h1>Heading 1</h1>
22       <p>Here's a small text description for the content. Nothing more, nothing less.</p>
23     </IonContent>
24   );
25 }
26 export default Example;

```

## 30 Toolbar

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

Kode 2.11: Contoh kode untuk membuat Toolbar

```

37 import React from 'react';
38 import { IonFooter, IonHeader, IonTitle, IonToolbar } from '@ionic/react';
39
40 function Example() {
41   return (
42     <>
43       <IonHeader>
44         <IonToolbar>
45           <IonTitle>Header Toolbar</IonTitle>
46         </IonToolbar>
47       </IonHeader>
48       <IonFooter>
49         <IonToolbar>
50           <IonTitle>Footer Toolbar</IonTitle>
51         </IonToolbar>
52       </IonFooter>
53     </>
54   );
55 }
56 export default Example;

```

### 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.

Pengembang yang menggunakan Capacitor harus menginstal capacitor tersebut terlebih dahulu dengan cara:

Kode 2.12: Kode untuk menginstal Capacitor Camera

```
$ npm install @capacitor/camera
```

Berikut cara menggunakan Capacitor dengan menggunakan plugin Camera:

Kode 2.13: Contoh kode Capacitor

```
import { Camera, CameraResultType } from '@capacitor/camera';

const takePicture = async () => {
  const image = await Camera.getPhoto({
    quality: 90,
    allowEditing: true,
    resultType: CameraResultType.Uri
  });
  const imageUrl = image.webPath;
  imageElement.src = imageUrl;
};
```

### Action Sheet

Action Sheet merupakan sebuah dialog yang menampilkan serangkaian opsi di atas konten aplikasi dan harus ditutup secara manual oleh pengguna sebelum mereka dapat melanjutkan interaksi dengan aplikasi. Opsi yang merusak dibuat jelas dalam mode ios. Ada beberapa cara untuk menutup action-sheet, termasuk mengetuk latar belakang atau menekan tombol escape di desktop. Action-sheet dapat dikendalikan menggunakan properti isOpen untuk mengontrol status presentasi dari Action Sheet dari state aplikasi. Action-sheet juga memiliki properti seperti header untuk judul, buttons untuk menambahkan tombol, dan keyboardClose untuk menutup action-sheet ketika tombol keyboard tertentu ditekan. Berikut cara menginstall Action Sheet:

Kode 2.14: Kode untuk menginstal Plugin Action Sheet

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

Berikut merupakan contoh dari penggunaan Action Sheet:

Kode 2.15: Contoh kode plugin Action Sheet

```
import { ActionSheet, ActionSheetButtonStyle } from '@capacitor/action-sheet';

const showActions = async () => {
  const result = await ActionSheet.showActions({
    title: 'Photo Options',
    message: 'Select an option to perform',
    options: [
      {
        title: 'Upload',
      },
      {
        title: 'Share',
      },
    ],
  });
};
```

```

13     },
14     {
15         title: 'Remove',
16         style: ActionSheetButtonStyle.Destructive,
17     },
18 ],
19 });
20
21 console.log('Action Sheet result:', result);
22 };

```

## 12 Camera

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

Kode 2.16: Contoh kode Capacitor Camera

```

19
20 1 import { Camera, CameraResultType } from '@capacitor/camera';
21 2
22 3 const takePicture = async () => {
23 4     const image = await Camera.getPhoto({
24 5         quality: 90,
25 6         allowEditing: true,
26 7         resultType: CameraResultType.Uri
27 8     });
28 9
29 0     // image.webPath will contain a path that can be set as an image src.
30 1     // You can access the original file using image.path, which can be
31 2     // passed to the Filesystem API to read the raw data of the image,
32 3     // if desired (or pass resultType: CameraResultType.Base64 to getPhoto)
33 4     var imageUrl = image.webPath;
34 5
35 6     // Can be set to the src of an image now
36 7     imageElement.src = imageUrl;
37 8 };
38

```

## 39 Filesystem

40 Filesystem API menyediakan alat NodeJS-like untuk bekerja dengan file pada perangkat. Pengem-  
 41 bang dapat menggunakan plugin ini untuk melakukan operasi file umum seperti membaca, tulis,  
 42 dan mengelola isi direktori.

43 Berikut kode untuk menginstall Filesystem:

Kode 2.17: Kode untuk menginstal plugin Filesystem

```

44
45 1 npm install @capacitor/action-sheet

```

47 Berikut contoh penggunaan Filesystem Plugin:

Kode 2.18: Contoh kode dari penggunaan Filesystem

```

48
49 1 import { Filesystem, Directory, Encoding } from '@capacitor/filesystem';
50 2
51 3 const writeSecretFile = async () => {
52 4     await Filesystem.writeFile({
53 5         path: 'secrets/text.txt',
54 6         data: 'This is a test',
55 7         directory: Directory.Documents,
56 8         encoding: Encoding.UTF8,
57 9     });

```

```

110 };
111
112 const readSecretFile = async () => {
113   const contents = await Filesystem.readFile({
114     path: 'secrets/text.txt',
115     directory: Directory.Documents,
116     encoding: Encoding.UTF8,
117   });
118
119   console.log('secrets:', contents);
120 };
121
122 const deleteSecretFile = async () => {
123   await Filesystem.deleteFile({
124     path: 'secrets/text.txt',
125     directory: Directory.Documents,
126   });
127 };
128
129 const readFilePath = async () => {
130   // Here's an example of reading a file with a full file path. Use this to
131   // read binary data (base64 encoded) from plugins that return File URIs, such as
132   // the Camera.
133   const contents = await Filesystem.readFile({
134     path: 'file:///var/mobile/Containers/Data/Application/22A433FD-D82D-4989-8BE6-9FC49DEA20BB/Documents/text.txt',
135   });
136
137   console.log('data:', contents);
138 };

```

## 31 Preference

32 Plugin Preference pada Ionic 7 adalah alat yang memungkinkan Anda untuk menyimpan data  
 33 sederhana dalam bentuk kunci/nilai yang dapat diakses secara bersamaan. Preferences API  
 34 menyediakan area penyimpanan data yang mendukung kunci/nilai untuk aplikasi Ionic. Beberapa  
 35 fitur utama dari Plugin Preference meliputi:

- 36 • Mengatur grup preferences: Preferences groups digunakan untuk mengatur kunci/nilai pairs.  
 37 Menggunakan nilai 'NativeStorage' memberikan kompatibilitas belakang dengan cordova-  
 38 plugin-nativestorage.
- 39 • Mengakses hasil preferences: Menggunakan getResult() method sehingga mendapatkan nilai  
 40 dari preferences yang terkait dengan kunci tertentu
- 41 • Menyimpan dan mengatur preferences: Menggunakan metode set() untuk menyimpan atau  
 42 mengatur nilai preferences

43 Untuk menggunakan Plugin Preference dalam aplikasi Ionic, perlu menginstal plugin Capacitor  
 44 Preferences dengan cara:

### Kode 2.19: Kode untuk menginstal plugin Preference

```

45
46 1 npm install @capacitor/preferences

```

48 Berikut merupakan contoh kode dari penggunaan Preference:

### Kode 2.20: Contoh kode dari plugin Preference

```

49
50 1 import { Preferences } from '@capacitor/preferences';
51 2
52 3 const setName = async () => {
53 4   await Preferences.set({
54 5     key: 'name',
55 6     value: 'Max',
56 7   });
57 8 };
58 9
59 10 const checkName = async () => {
60 11   const { value } = await Preferences.get({ key: 'name' });

```

```
12 |  
23 |   console.log('Hello ${value}!');  
34 | };  
45 |  
56 | const removeName = async () => {  
67 |   await Preferences.remove({ key: 'name' });  
78 | };
```



## DAFTAR REFERENSI





# LAMPIRAN A

## KODE PROGRAM

Kode A.1: MyCode.c

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

Kode A.2: MyCode.java

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



## LAMPIRAN B

### HASIL EKSPERIMEN

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



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4