

SKRIPSI

PEMBUATAN ULANG APLIKASI RUGBY INDONESIA DENGAN IONIC 7 DAN CAPACITOR



Nathanael Adi Trianto

NPM: 6181901041

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

DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	v
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 Rugby Indonesia App	5
2.2 ReactJS	5
2.3 Ionic 7 Framework	6
2.3.1 UI Components	6
DAFTAR REFERENSI	9
A KODE PROGRAM	11
B HASIL EKSPERIMEN	13

DAFTAR GAMBAR

1.1	Halaman aplikasi Rugby Indonesia	2
B.1	Hasil 1	13
B.2	Hasil 2	13
B.3	Hasil 3	13
B.4	Hasil 4	13

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.¹ Olahraga ini dapat dimainkan dengan tangan dan juga tendangan, tetapi pemain hanya boleh melempar bola atau diserahkan ke belakang saat dibawa menggunakan tangan.²

Persatuan Rugby Union Indonesia (PRUI) adalah organisasi yang bertanggung jawab atas pengelolaan dan pengembangan Rugby Union di Indonesia.³ Mereka memiliki tim nasional putra dan putri.⁴ 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.⁵ Rugby union di Indonesia adalah olahraga minor namun berkembang, yang telah ada selama beberapa dekade, dan mengalami fluktuasi dalam kesuksesannya.⁶ Pada tahun 2023, terdapat 35 klub rugby, 700 pemain terdaftar, 45 pelatih terdaftar, dan 23 wasit terdaftar di Indonesia.⁷

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

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

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

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

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

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

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

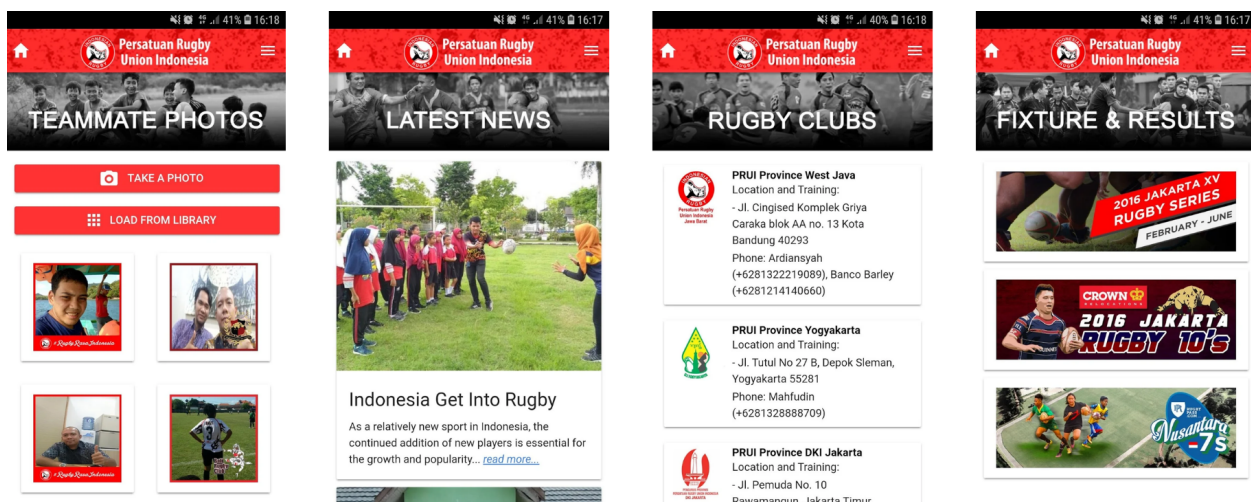
⁷<https://rugbyindonesia.or.id/>

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

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

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

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



Gambar 1.1: Halaman-halaman dari aplikasi Rugby Indonesia

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

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

⁹<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.
2. Pengguna hanya bisa mengunggah foto dan melihat foto unggahan dari pengguna lain, pengguna tidak dapat menghapus ataupun mengubah foto 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. Membangun aplikasi Rugby Indonesia yang sudah menggunakan *framework* Ionic 7 serta Capacitor.
4. Melakukan pengujian dan eksperimen.

5. 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 rakan-rakan 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. Aplikasi ini juga tersedia di GitHub dengan menggunakan framework ionic versi 2 untuk versi iOS.

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. 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 kontrol formulir seperti Toggle atau Input. Komponen Item dan Label tidak lagi diperlukan, dan setiap kontrol formulir 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.

Pengembang dapat mengikuti Panduan Migrasi Ionic 7 untuk memperbarui aplikasi Ionic 6 yang sudah ada. Pada saat tugas akhir ini dibuat, 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.

Action Sheet

Action Sheet dari Ionic atau yang bisa disebut juga 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.

1 **Accordion**

2 Accordion berfungsi untuk mengurangi ruang vertikal dalam mengorganisir informasi yang ingin
3 ditampilkan. Accordion memiliki 2 komponen yaitu ion-accordion dan ion-accordion-group. Ketika
4 menggunakan accordion, ion-accordion harus berada di dalam ion-accordion-group.

5 **Alert**

6 Alert pada Ionic berfungsi untuk memberikan informasi serta mengumpulkan informasi dari pengguna
7 menggunakan input dari pengguna. Sama seperti Action Sheet, Alert juga biasanya dimunculkan
8 di atas konten dari aplikasi, namun Alert biasanya berada di tengah konten aplikasi, sedangkan
9 Action Sheet muncul berada dari bawah aplikasi.

DAFTAR REFERENSI

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

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