



Media Handling

Chapter 5 - Topic 1

**Selamat datang di Chapter 5 Topic 1 online course
React Native dari Binar Academy!**





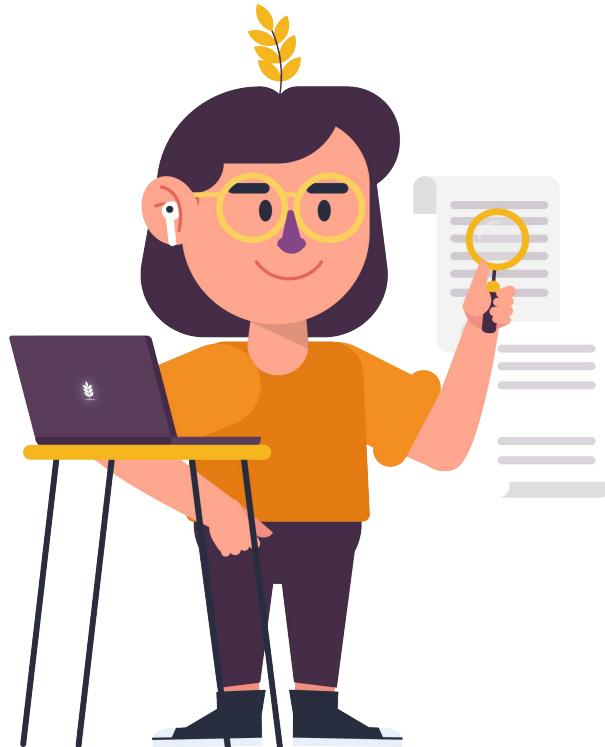
Hei hei~

Sudah masuk Chapter 5 nih! Semoga kamu sudah paham chapters sebelumnya dan siap lanjut dengan materi selanjutnya.

Oke, di Chapter 4 kita sudah belajar tentang Redux, nah di Chapter 5 Topic 1 ini kamu akan belajar tentang **media handling**. Bukan.. bukan media handling kayak memanggil media massa buat press conference artis-artis hehe..

Hmm, jadi media handling kayak apa ya?

Yuk langsung cari tahu di materi ini!



Detailnya, kita bakal bahas hal-hal berikut ini:

- Sumber media
- Cara menampilkan video pada React Native
- Memahami cara menampilkan dokumen (PDF) pada React Native





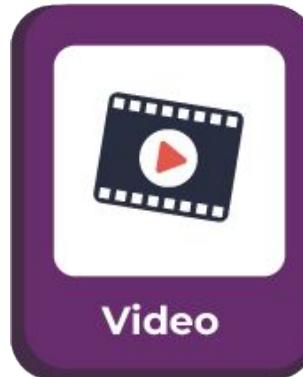
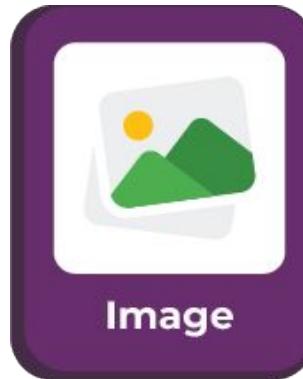
Kalau kamu dengar kata media, mungkin yang terbayang adalah media cetak atau elektronik kayak koran, majalah, atau televisi, tapi yang akan kita bahas bukan itu, guys.





Kalau kamu perhatiin, sekarang ada banyak sekali aplikasi yang pakai media sebagai salah satu kontennya.

Media yang di maksud di sini berupa **video, image, audio, dan dokumen PDF**.





Contohnya, aplikasi YouTube pakai media video sebagai konten utama mereka.

Aplikasi banking, seperti Jenius menampilkan dokumen pdf sebagai bukti transaksi.

Aplikasi Shopee yang selalu memunculkan audio dengan suara 'Shopeee' setiap kali ada notifikasi masuk.





Apa saja sih sumber media?

Nah, media pada dasarnya memiliki dua sumber, guys, yaitu lokal dan remote.

Hmm.. apa bedanya ya?





Kalau media lokal...

Media yang bersumber dari lokal maksudnya adalah, sebuah **file yang dimasukan ke dalam sebuah project aplikasi.**

Contohnya, kamu punya gambar kucing yang ingin kamu tampilkan sebagai background dari aplikasimu, lalu gambar tersebut kamu copy/masukkan ke dalam folder project secara manual sebagai bagian dari asset yang nantinya akan ditampilkan pada aplikasimu .





Ada beberapa kekurangan dan kelebihan menggunakan media yang bersumber dari lokal, yaitu:



Kelebihan

- Lebih cepat di-load oleh aplikasi
- Nggak perlu di-load dari server
- Media tetap muncul meski tanpa ada koneksi internet

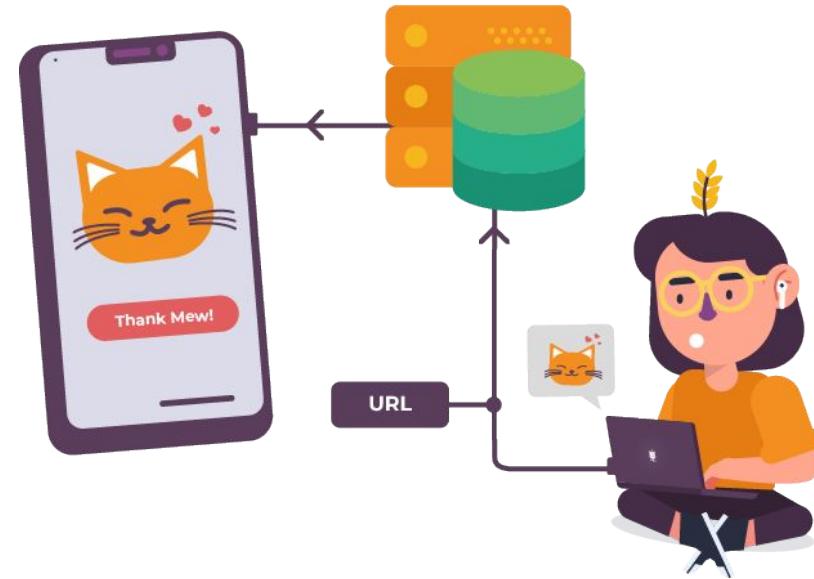
Kekurangan

- Konten media nggak bisa berubah-ubah
- Akan menambah ukuran bundle aplikasi menjadi lebih besar



Sedangkan media remote..

Adalah **media yang bersumber dari remote atau dari server**. Artinya file media ini nggak disimpan di folder project, tapi disimpan di server.

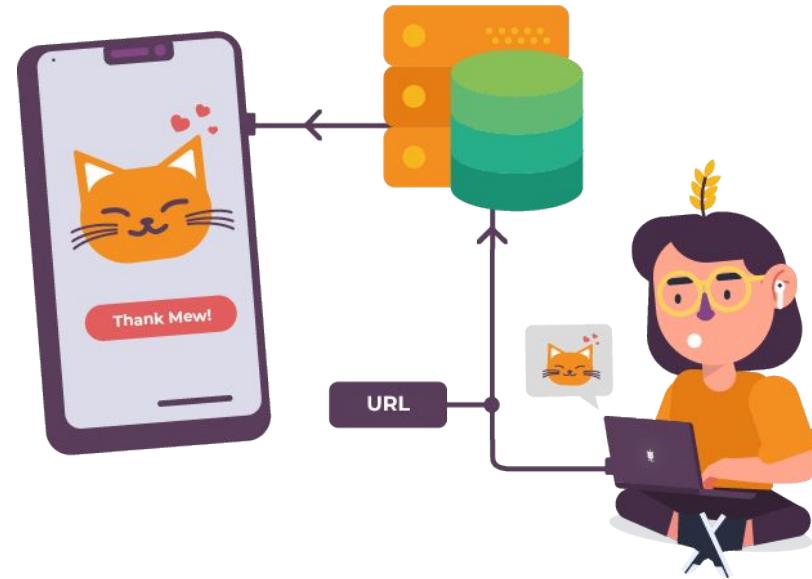




Terus bagaimana ya cara mengakses media remote?

Media yang disimpan di server (remote) dapat di akses **melalui sebuah URL**. Nah, contoh url untuk mengakses media dokumen yang terletak pada sebuah server kayak gini ya!

<http://samples.leanpub.com/theractnativebook-sample.pdf>





Inilah beberapa kekurangan dan kelebihan kalau kamu menggunakan media yang bersumber dari remote:



Remote / Server

Kelebihan

- Konten media dinamis, dapat berubah-ubah, contohnya video yang ditampilkan YouTube di halaman home yang selalu berubah-ubah.
- Nggak menambah ukuran bundle aplikasi, karena file media tersebut disimpannya di server.

Kekurangan

- Konten media nggak bisa diakses kalau aplikasi nggak terkoneksi ke internet atau server tempat penyimpanan media mati.
- Loading aplikasi lebih lama dan bergantung dengan kecepatan koneksi internet user.



Nah, itulah perbedaan antara media lokal dan remote. Mana yang lebih kamu suka?

Manapun yang kamu pilih, penggunaannya bisa kamu sesuaikan dengan kondisi dan kebutuhanmu ya.





Oke, kamu sudah tahu perbedaan kedua media ya. Sekarang, kita akan membahas cara menampilkan media yang bersumber dari **remote (server)**.

Let's go!





Rendering Video

Aplikasi mobile umumnya hanya menampilkan konten dalam bentuk video untuk memberikan informasi yang lebih jelas dan interaktif kepada user.

FYI, rendering video ini maksudnya adalah memasukkan video ke dalam aplikasimu ya, guys!





Seperti contoh aplikasi TikTok yang menampilkan konten berupa video agar penggunanya lebih terhibur dan nyaman.

Aplikasi TikTok adalah salah satu contoh **penerapan rendering video yang bersumber dari remote(server)**.





Lalu gimana sih prosesnya di Tiktok?

Di awal kita udah tahu nih kalo sumbe media itu ada yang disimpan di local dan di server. Nah, video-video yang ada di Tiktok itu disimpannya di server, guys!

Jadi kalau mau menampilkan video itu ke user di aplikasi TikTok, kita harus memasukkan link videonya di source code, contohnya <https://video.mp4>.





Oke, sekarang kita coba menampilkan video yang berasal dari server pada Aplikasi React Native kita, yuk.





1. Persiapkan project React Native

Kamu bisa pakai Project yang sudah ada atau kamu bisa bikin project React Native baru khusus untuk belajar video.

Kita akan coba membuat project React Native baru.



```
npx react-native init rnmedia
```

2. Install react-native-video

React-native-video adalah sebuah package/library untuk menampilkan video pada React Native.

Package ini menawarkan beberapa fungsi seperti untuk mute video, menampilkan video, pause video, play video dan play audio saja.





Untuk menginstall package ini, ketik command berikut pada terminal:

```
npm install react-native-video
```

Setelah proses instalasi selesai, khusus iOS, kamu perlu menginstall package pods nya juga agar instalasi berjalan lancar.

Ketikan perintah **cd ios && pod install**

```
● ● ●  
npm install react-native-video
```

```
● ● ●  
cd ios && pod install
```



3. Edit file App.js

Selanjutnya, edit file App.js yang terdapat pada root folder. Edit file tersebut seperti syntax pada kode di bawah ini.

```
import React from 'react';
import {View, Text} from 'react-native';

class App extends React.Component {
    render() {
        return (
            <View style={{flex: 1, alignItems: 'center', justifyContent: 'center'}}>
                <Text>My Video project!</Text>
            </View>
        );
    }
}

export default App;
```



4. Running App

Jalankan aplikasi React Native nya dengan command:

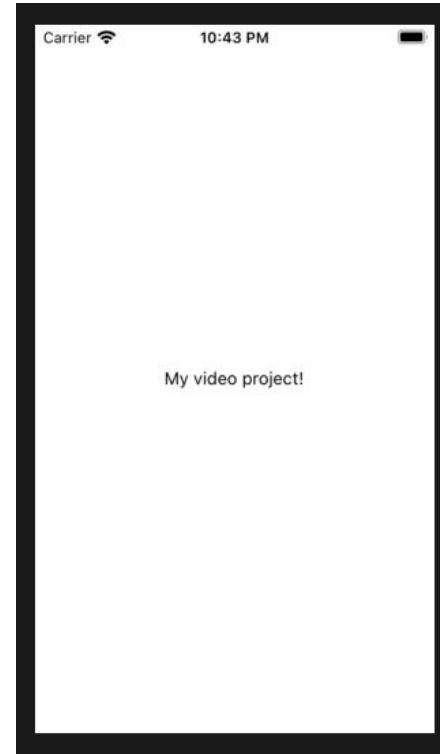
Untuk menjalankan Android:

```
npm run android
```

Untuk menjalankan iOS:

```
npm run ios
```

Setelah dijalankan, maka tampilan aplikasi kamu akan persis seperti gambar di samping.



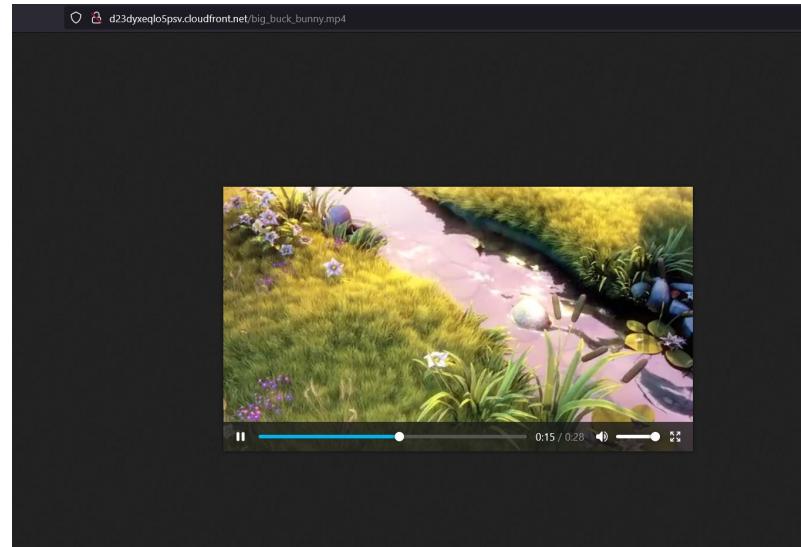


5. Sumber Video

Sekarang kamu akan menampilkan video yang ada di server. Jadi URL video yang akan kamu gunakan yaitu:

http://d23dyxeqlo5psv.cloudfront.net/big_buck_bunny.mp4

Video itulah yang akan kita tampilkan pada React Native.





6. Import Video Component

Langkah selanjutnya adalah import video component yang berasal dari package react-native-video

```
import React from 'react';
import {View, Text} from 'react-native';

import Video from 'react-native-video';

class App extends React.Component {
  render() {
    return (
      <View style={{alignItems: 'center', flex: 1, justifyContent: 'center'}>
        <Text>My video project!</Text>
      </View>
    );
  }
}

export default App;
```



7. Panggil component Video

Langkah selanjutnya, panggil component Video dengan memasukan URL video pada bagian props source. Di dalam props source ini kamu harus memasukkan alamat URL video nya sebagai value dari property "uri".

Selain itu, berikan component `<Video />` sebuah styling dengan mendefinisikan kode-kode styling pada props style yang terdapat component `<Video />`

```
import React from 'react';
import {View, Text} from 'react-native';

import Video from 'react-native-video';

class App extends React.Component {
  render() {
    return (
      <View style={{alignItems: 'center', flex: 1, justifyContent: 'center'}}>
        <Text>My video project!</Text>
        <Video
          source={{
            uri: 'http://d23dyxeqlo5psv.cloudfront.net/big_buck_bunny.mp4',
          }}
          style={{
            width: 300,
            height: 300,
          }}
        />
      </View>
    );
  }
}

export default App;
```

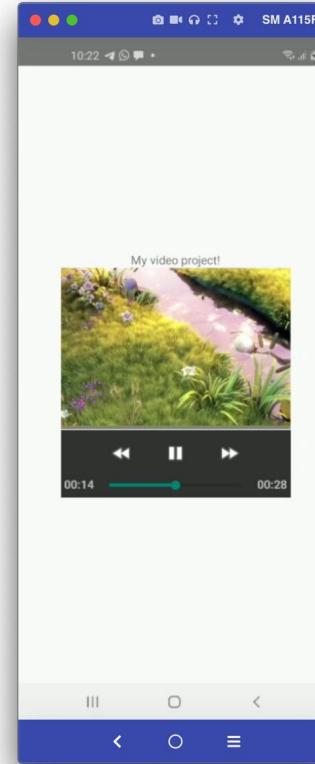


8. Restart aplikasi React Native

Restart ulang aplikasi React Native kamu, maka akan muncul sebuah tampilan video pada screen.

Ukuran tampilan video bisa kamu sesuaikan pada width dan height di props style yang terdapat pada component <Video />

Nah, sekarang video nya udah dapat di play di aplikasi react native kamu deh, guys.





9. Menambahkan Thumbnails/Poster

Nah, sekarang kamu perlu menambahkan poster/thumbnail. Jadi ketika videonya masih dalam proses loading, kamu akan memunculkan sebuah gambar default, kira-kira kayak thumbnail di YouTube.

Yuk, silakan coba munculkan thumbnail dengan link gambar di samping.



Link gambar:

https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/English_Cocker_Spaniel_4.jpg/800px-English_Cocker_Spaniel_4.jpg



Masih lanjut tentang menambahkan Thumbnails/Poster~

Kalau melihat dokumentasi library react-native-video, untuk menampilkan thumbnail, kamu juga bisa menambahkan props poster dengan value URL image yang akan ditampilkan menjadi poster.

https://github.com/react-native-video/react-native-video#poster

- Set the ignoreSilentSwitch prop to "ignore"

Platforms: Android ExoPlayer, Android MediaPlayer, iOS

playWhenInactive

Determine whether the media should continue playing when notifications or the Control Center are in front of the video.

- **false (default)** - Don't continue playing the media
- **true** - Continue playing the media

Platforms: iOS

poster

An image to display while the video is loading

Value: string with a URL for the poster, e.g. "<https://baconmockup.com/300/200/>"

Platforms: all

posterResizeMode

Determines how to resize the poster image when the frame doesn't match the raw video dimensions.

- **"contain" (default)** - Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or less than the corresponding dimension of the view (minus padding).
- **"center"** - Center the image in the view along both dimensions. If the image is larger than the view, scale it down uniformly so that it is contained in the view.
- **"cover"** - Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or larger than the corresponding dimension of the view (minus padding).

Link:

<https://www.npmjs.com/package/react-native-video#poster>



Sekarang tambahkan props poster dengan value url image:
https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/English_Cocker_Spaniel_4.jpg/800px-English_Cocker_Spaniel_4.jpg

Nah kemudian save dan refresh aplikasi react native kamu.

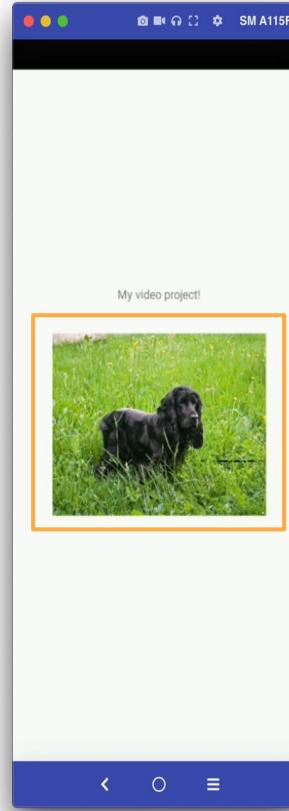
```
import React from 'react';
import {View, Text} from 'react-native';

import Video from 'react-native-video';
class App extends React.Component {
  render() {
    return (
      <View style={{alignItems: 'center', flex: 1, justifyContent: 'center'}}>
        <Text>My video project!</Text>
        <video
          source={{
            uri: 'http://d23dyxeqlo5psv.cloudfront.net/big_buck_bunny.mp4',
          }}
          style={{
            width: 300,
            height: 300,
          }}
          fullscreen={true}
          controls={true}
          poster="https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/English_Cocker_Spaniel_4.jpg/800px-English_Cocker_Spaniel_4.jpg"
        />
      </View>
    );
  }
}

export default App;
```



Sekarang kalau kamu buka ulang aplikasi React Native-mu, maka yang akan muncul pertama kali adalah gambar poster/thumbnil terlebih dahulu, baru setelah proses loading videonya selesai, video akan langsung muncul secara otomatis.

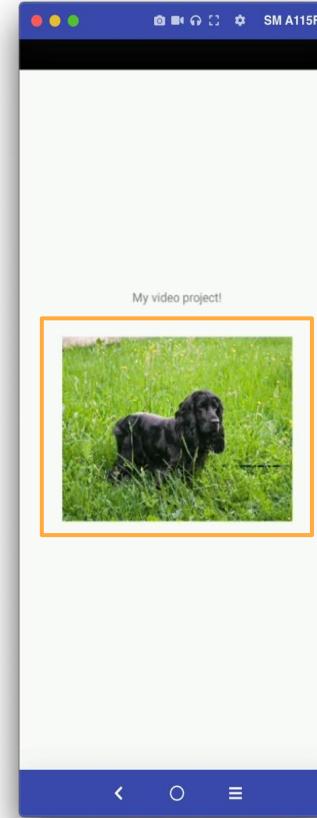




So that's it guys!

Sekarang kamu sudah bisa menampilkan video pada aplikasi React Native-mu.

Ada banyak hal yang bisa kamu eksplor terkait video, untuk detailnya silahkan eksperimen dan eksplor dokumentasi [di sini](#) ya!





Setelah belajar tentang media yang bersumber dari remote, sekarang kita lanjut ke pembahasan tentang **Dokumen** ya.

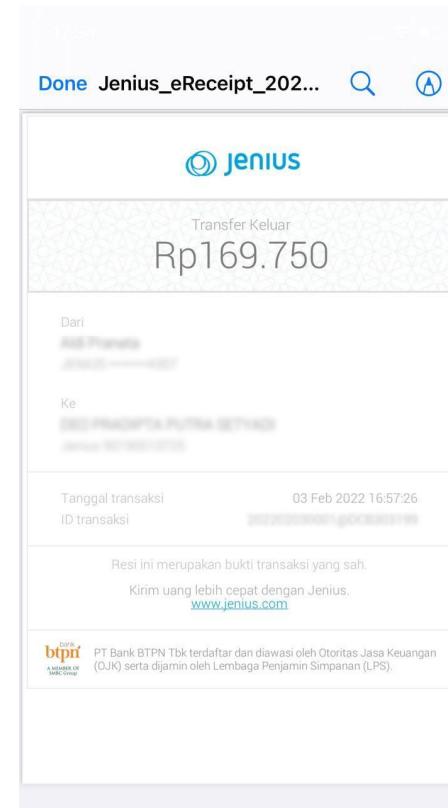




Format Dokumen yang umum digunakan adalah PDF!

Elemen lain dalam media adalah dokumen PDF. FYI, aplikasi mobile umumnya hanya menampilkan dokumen dalam format PDF saja.

Misalnya saat kamu transfer uang atau pembayaran menggunakan aplikasi Jenius, maka pada aplikasi Jenius kamu dapat melihat dokumen bukti transaksi tersebut dalam format PDF.



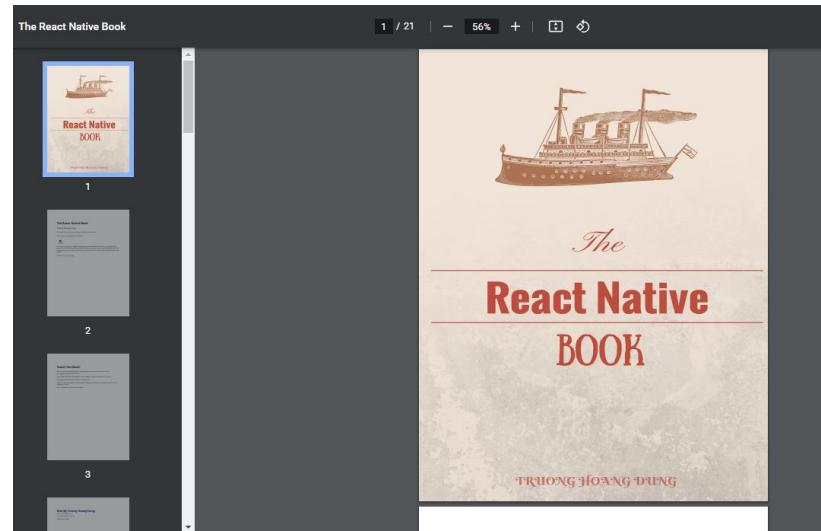


Terus, bagaimana ya proses rendering dokumen (PDF)?

Oke, sekarang kamu akan coba belajar merender alias menampilkan dokumen PDF pada screen React Native-mu.

Alamat URL PDF yang akan kamu tampilkan dapat diakses pada:

<http://samples.leanpub.com/theractnativebook-sample.pdf>





Nah, yang perlu kamu lakukan pertama adalah install React Native PDF.

Berikut langkah-langkahnya, guys:

1. Gunakan sebuah library/package yang bernama react-native-pdf.

Untuk menginstall react-native-pdf pada aplikasi, ketik perintah:

```
npm install react-native-pdf react-native-blob-utils fbjs
```

```
#pake npm
npm install react-native-pdf react-native-blob-utils fbjs

#pake yarn
yarn add react-native-pdf react-native-blob-utils fbjs
```



2. Setelah itu buka file android/app/build.gradle.
3. Tambahkan script tersebut dalam object android yang ada pada file android/app/build.gradle.
4. Setelah ditambahkan, file android/app/build.gradle akan tampak seperti gambar kedua di samping.

```
## SCRIPT YANG AKAN KAMU TAMBAHKAN
packagingOptions {
    pickFirst 'lib/x86/libc++_shared.so'
    pickFirst 'lib/x86_64/libjsc.so'
    pickFirst 'lib/arm64-v8a/libjsc.so'
    pickFirst 'lib/arm64-v8a/libc++_shared.so'
    pickFirst 'lib/x86_64/libc++_shared.so'
    pickFirst 'lib/armeabi-v7a/libc++_shared.so'
}

120  /*
121  def enableHermes = project.ext.react.get("enableHermes", false);
122
123  /**
124   * Architectures to build native code for in debug.
125   */
126  def nativeArchitectures = project.getProperties().get("reactNativeDebugArchitectures")
127
128  android {
129
130      packagingOptions { ... You, 10 seconds ago • Uncommitted changes
131          pickFirst 'lib/x86/libc++_shared.so'
132          pickFirst 'lib/x86_64/libjsc.so'
133          pickFirst 'lib/arm64-v8a/libjsc.so'
134          pickFirst 'lib/arm64-v8a/libc++_shared.so'
135          pickFirst 'lib/x86_64/libc++_shared.so'
136          pickFirst 'lib/armeabi-v7a/libc++_shared.so'
137          ...
138
139      ndkVersion rootProject.ext.ndkVersion
140
141      compileSdkVersion rootProject.ext.compileSdkVersion
142
143      defaultConfig {
144          applicationId "com.mytodoapp"
145          minSdkVersion rootProject.ext.minSdkVersion
146          targetSdkVersion rootProject.ext.targetSdkVersion
147          versionCode 1
148          versionName "1.0"
149      }
}
```



Langkah kedua, membuat tampilan~

Pada file app.js yang terdapat pada root project, edit component App tersebut sesuai dengan syntax kode pada gambar di samping

```
import React from 'react';
import {View, Text, Button} from 'react-native';

class App extends React.Component {

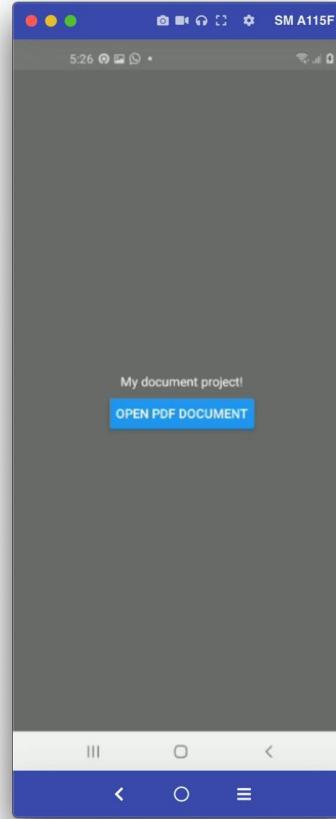
  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />
      </View>
    );
  }
}

export default App;
```



Nah, saat kamu refresh aplikasi React Native, akan muncul tampilan layout seperti gambar di samping.

Selanjutnya, kamu akan menampilkan PDF-nya dengan component `<PDF />` dari `react-native-pdf`





Terakhir, saatnya mengimplementasikan react-native-pdf

Bagaimana caranya? Begini:

1. Import component PDF dari package react-native-pdf dengan syntax:

```
import Pdf from 'react-native-pdf';
```

```
import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />
      </View>
    );
  }

  export default App;
```



2. Kemudian, panggil component PDF tersebut dengan memasukan props source.

Nah, di props source inilah nantinya kamu akan memasukkan link URL file PDF yang ingin kamu tampilkan di dalam property URL.

```
● ● ●  
import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Pdf
          source={{
            uri: 'http://samples.leanpub.com/theractnativebook-sample.pdf',
          }}
        />
      </View>
    );
  }

  export default App;
```



3. Karena kamu memberi style ukuran width 300 dan height 300, maka tampilan PDF akan mengikuti dimensi 300x300.

Sekarang coba ubah dimensinya jadi 600 x 600, width 600 kemudian height nya 600.





- Setelah itu, pada komponen PDF tambahkan juga props style. Di props style ini kamu bisa mengatur seberapa besar document PDF ini akan tampil pada screen aplikasi React Native-mu.

Sekarang masukan width 300 dan height 300.

```
import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />

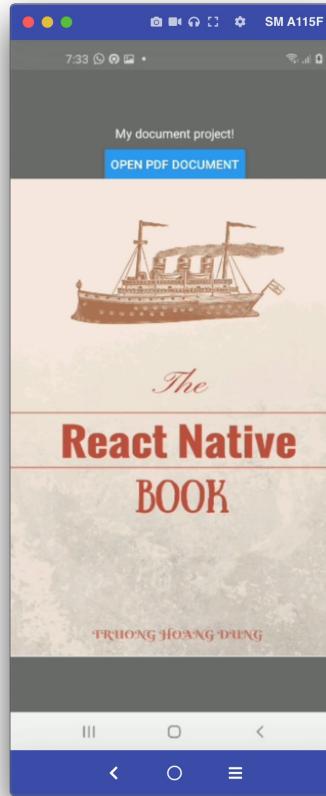
        <Pdf
          source={{
            uri: 'http://samples.leanpub.com/theractnativebook-sample.pdf',
          }}
          style={{height: 300, width: 300}}
        />
      </View>
    );
  }
}

export default App;
```



Nah berhasil kan?

Sekarang tampilan PDF nya menjadi lebih besar.





5. Supaya tampilan lebih rapi, berikan margin Top sebesar 10 pada style yang terdapat di component <PDF />. Tujuannya agar ada jarak antara PDF dengan button yang berwarna biru

```
import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

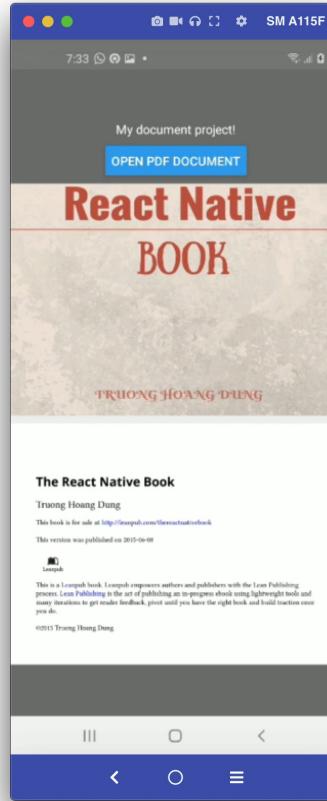
  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: 'ffff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />

        <Pdf
          source={{
            uri: 'http://samples.leanpub.com/theractnativebook-sample.pdf',
          }}
          style={{height: 600, width: 600, marginTop: 10}}
        />
      </View>
    );
  }

  export default App;
```



Sekarang sudah terlihat jarak antara PDF dengan button berwarna biru. Jadi kelihatan lebih rapi yaa.





By the way guys, saat aplikasi pertama kali dibuka, dokumen PDF kan akan langsung muncul.

Jadi, kita coba bikin Document PDF-nya hanya tampil saat user menekan tombol “Open PDF Document” saja, yuk.





Langkah pertama: Buat state pdfLink

Yak! Buat sebuah state dengan nama pdfLink. State pdfLink ini nantinya akan **berisi link PDF yang akan ditampilkan pada screen.**

Beri initial value pada state tersebut string kosong ("").

```
● ● ●

import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  state = {
    pdfLink: ''
  }

  render() {
    return (
      <View
        style={{{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />

        <Pdf
          source={{
            uri: 'http://samples.leanpub.com/theractnativebook-sample.pdf',
          }}
          style={{height: 600, width: 600, marginTop: 10}}
        />
      </View>
    );
  }
}

export default App;
```



Kemudian pada bagian props source yang terdapat pada komponen <Pdf />, gantilah value dari properti URL menjadi variabel state pdfLink yang telah kamu buat.

```
● ● ●

import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

    state = {
        pdfLink: ''
    }

    render() {
        return (
            <View
                style={{
                    alignItems: 'center',
                    flex: 1,
                    justifyContent: 'center',
                    backgroundColor: '#696969',
                }}>
                <Text style={{color: 'fff', marginBottom: 10}}>
                    My document project!
                </Text>
                <Button title="Open PDF Document" />

                <Pdf
                    source={{
                        uri: this.state.pdfLink,
                    }}
                    style={{height: 600, width: 600, marginTop: 10}}
                />
            </View>
        );
    }

    export default App;
```



Langkah selanjutnya, buatlah conditional render terhadap variable `this.state.pdfLink`.

Kalau state `pdfLink` isinya masih string kosong, kamu nggak perlu menampilkan apa-apa, sebaliknya, kalau state `pdfLink` sudah terisi URL, maka kamu perlu menampilkan komponen PDF-nya.

```
import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  state = {
    pdfLink: ''
  }

  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />
        {this.state.pdfLink ? <Pdf
          source={{
            uri: this.state.pdfLink,
          }}
          style={{height: 600, width: 600, marginTop: 10}}
        /> : null }
      </View>
    );
  }
}

export default App;
```



Langkah kedua: Function onButtonOpenClick

Buat sebuah method function dengan nama onButtonOpenClick. Saat function ini di-trigger, maka function ini akan mengupdate value state pdfLink dengan URL PDF:

'<http://samples.leanpub.com/thereactnativebook-sample.pdf>'.

```
● ● ●

import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  state = {
    pdfLink: ''
  }

  onButtonOpenClick = () => {
    this.setState({
      pdfLink: 'http://samples.leanpub.com/thereactnativebook-sample.pdf',
    });
  }

  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}>
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" />
        {this.state.pdfLink ? <Pdf
          source={{
            uri: this.state.pdfLink,
          }}
          style={{height: 600, width: 600, marginTop: 10}}
        /> : null }
      </View>
    );
  }
}

export default App;
```



Langkah terakhir, masukan function onButtonOpenClick yang telah kamu buat sebelumnya pada props onPress di komponen <Button />.

Function onButtonOpenClick akan di-trigger saat user mengklik tombol 'Open PDF Document'.

```
● ● ●

import React from 'react';
import {View, Text, Button} from 'react-native';

import Pdf from 'react-native-pdf';

class App extends React.Component {

  state = {
    pdfLink: ''
  }

  onButtonOpenClick = () => {
    this.setState({
      pdfLink: 'http://samples.leanpub.com/theractnativebook-sample.pdf',
    });
  };

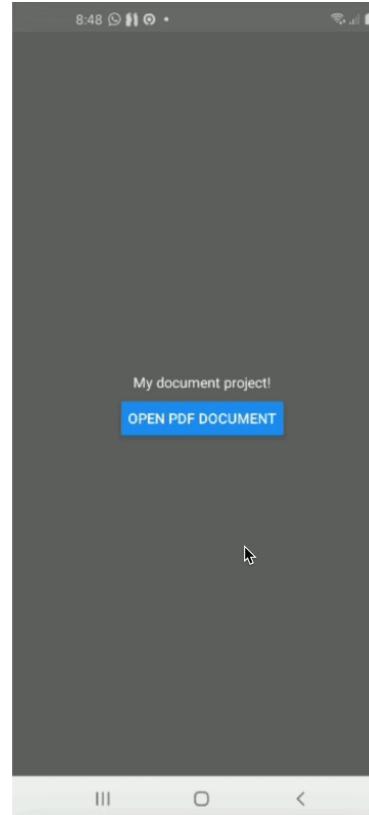
  render() {
    return (
      <View
        style={{
          alignItems: 'center',
          flex: 1,
          justifyContent: 'center',
          backgroundColor: '#696969',
        }}
      >
        <Text style={{color: '#fff', marginBottom: 10}}>
          My document project!
        </Text>
        <Button title="Open PDF Document" onPress={this.onButtonOpenClick}>
          {this.state.pdfLink ? <Pdf
            source={{
              uri: this.state.pdfLink,
            }}
            style={{height: 600, width: 600, marginTop: 10}}
          /> : null}
        </View>
      );
    }
}

export default App;
```



That's it!

Nah, sekarang kamu sudah berhasil deh menampilkan PDF pada screen aplikasi React Native kamu, guys!



Saatnya kita Quiz!





1. Apa kekurangan bila menyimpan media (Video, gambar, dokumen) dilocal?

- A. Akan menambah ukuran bundle aplikasi.
- B. Akan rentan terhadap keamanan aplikasi.
- C. Kualitas media yang ditampilkan akan menurun.



2. Pernyataan dibawah ini yang benar adalah..

- A. React native hanya dapat menampilkan media yang bersumber dari server saja.
- B. React native hanya dapat menampilkan media yang bersumber dari lokal saja, tidak bisa dari remote, karena security issue.
- C. React native dapat menampilkan media yang bersumber dari server dan lokal.



3. Apa fungsi dari perintah `npx react-native link <NAMA_PACKAGE>`?

- A. Untuk menghubungkan native dependencies suatu package terhadap aplikasi react native
- B. Untuk membuat link koneksi antara react native dan library
- C. Untuk menghubungkan package/library terhadap server npm



4. Di bawah ini yang merupakan kegunaan untuk melakukan custom configuration terhadap suatu package yang kita pakai adalah...

- A. Configurable Props
- B. npm install <Packages>
- C. pod install

Terima Kasih!



Next Topic

loading...