**PRAKTIKUM**

**PENGEMBANGAN APLIKASI BERGERAK**

**MODUL 3**

**React Native Core Component**

Disusun oleh:

**ZULFA AZKA FARISADILAH**

**1204220109**

****

**PROGRAM STUDI SARJANA SISTEM INFORMASI**

**DIREKTORAT KAMPUS SURABAYA**

**UNIVERSITAS TELKOM**

**SURABAYA**

**2025**

**DAFTAR ISI**

[**1** **Membuat & Menjalankan Project React Native via Expo** 3](#_Toc211766635)

[**2** **Persiapan Struktur Project** 5](#_Toc211766636)

[**3** **Inisialisasi App.js** 5](#_Toc211766637)

[**4** **Komponen Header** 6](#_Toc211766638)

[**5** **Komponen Button** 7](#_Toc211766639)

[**6** **Komponen Separator** 8](#_Toc211766640)

[**7** **Screen List** 9](#_Toc211766641)

[**8** **Screen Article** 11](#_Toc211766642)

[**9** **Output Keseluruhan** 12](#_Toc211766643)

[**10** **Tugas** 15](#_Toc211766644)

[**11** **Link Repository GitHub:** 16](#_Toc211766645)

# **Membuat & Menjalankan Project React Native via Expo**

**Input:**

* Buat sebuah project **React Native** via **Expo** bernama **basic-core-component** dengan menjalankan command:
  + **npx create-expo-app basic-core-component --template blank**
* Masuk ke folder project **basic-core-component** yang sudah terbuat dengan menjalankan command:
  + **cd basic-core-component**
* Jalankan project react native anda dengan command:
  + **npx expo start** atau
  + **npx expo start --tunnel**
* Akan muncul sebuah **QRCode** pada command prompt anda

Buka aplikasi **Expo Go** di handphone, kemudian scan **QRCode** tersebut.

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

A white background with black text

AI-generated content may be incorrect.

# **Persiapan Struktur Project**

A screenshot of a computer

AI-generated content may be incorrect.

# **Inisialisasi App.js**

**Input:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Penjelasan:**

Bagian pertama melakukan import beberapa modul React Native seperti useState, useRef, View, DrawerLayoutAndroid, dan StatusBar, serta beberapa komponen buatan sendiri seperti Header, Button, Separator, List, dan Article.

Di dalam komponen App, dibuat sebuah state bernama page menggunakan useState("list"). Nilai awalnya adalah "list", artinya saat pertama kali aplikasi dijalankan, yang tampil adalah halaman daftar (List). Kemudian ada juga ref bernama drawer yang digunakan untuk mengontrol navigasi geser (Drawer).

Fungsi changePage dibuat untuk mengubah halaman sesuai tombol yang ditekan. Saat tombol ditekan, laci navigasi (Drawer) akan ditutup melalui drawer.current.closeDrawer(), kemudian state page diubah sesuai halaman tujuan.

Fungsi navigationView berisi tampilan menu navigasi di dalam drawer. Isinya terdiri dari tiga tombol: tombol “List” untuk menuju halaman daftar, tombol “Article” untuk menuju halaman artikel, dan tombol “Close” untuk menutup drawer tanpa berpindah halaman.

Bagian return menampilkan struktur utama aplikasi. Komponen DrawerLayoutAndroid digunakan untuk menampilkan laci navigasi di sisi kiri layar, dengan lebar 300 piksel dan tampilan menu dihasilkan oleh navigationView. Di dalamnya terdapat StatusBar yang mengatur warna bilah status, komponen Header di bagian atas, dan area konten yang menampilkan halaman sesuai nilai state page, apakah itu List atau Article.

# **Komponen Header**

**Input:**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Penjelasan:**

Bagian pertama melakukan import beberapa elemen bawaan dari React Native, yaitu View, TouchableOpacity, Image, dan StyleSheet. Elemen-elemen ini digunakan untuk membuat tampilan, tombol yang bisa ditekan, menampilkan gambar, dan menata gaya (style).

Selanjutnya dibuat sebuah komponen fungsional bernama Header yang menerima props (parameter dari komponen lain). Di dalamnya terdapat struktur tampilan menggunakan tag <View>. Pada bagian kiri header, terdapat sebuah tombol (TouchableOpacity) yang berisi ikon menu (menu.png). Ketika tombol ini ditekan, fungsi onPress akan memanggil props.drawer.current.openDrawer(), yang berarti membuka navigasi geser (drawer) dari aplikasi utama.

Pada bagian kanan header, terdapat sekumpulan ikon media sosial seperti Facebook, YouTube, Twitter, dan ikon pencarian. Semua ikon tersebut ditampilkan dalam satu baris menggunakan gaya iconsView dengan pengaturan arah horizontal (flexDirection: "row") dan posisi yang sejajar di tengah (alignItems: "center").

Bagian akhir kode berisi objek styles yang diatur menggunakan StyleSheet.create().

* header: mengatur warna latar belakang merah tua (#AA0002), tata letak horizontal, jarak antar elemen (justifyContent: "space-between"), dan padding 15 piksel.
* iconsView: memastikan semua ikon tersusun sejajar di tengah.
* icons: mengatur ukuran setiap ikon (lebar 36, tinggi 16) dan menyesuaikan proporsi gambar agar tidak terpotong dengan resizeMode: "contain".

Terakhir, komponen Header diekspor menggunakan export default Header, sehingga bisa digunakan di file lain seperti App.js.

# **Komponen Button**

**Input:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Penjelasan:**

Baris pertama melakukan import elemen bawaan dari React Native, yaitu TouchableOpacity, Text, dan StyleSheet.

* TouchableOpacity berfungsi membuat area yang dapat ditekan seperti tombol.
* Text digunakan untuk menampilkan tulisan di dalam tombol.
* StyleSheet dipakai untuk menata gaya tampilan tombol agar rapi dan konsisten.

Selanjutnya, dibuat komponen fungsional bernama Button yang menerima parameter props. Parameter ini digunakan agar tombol bisa menampilkan teks dan fungsi berbeda sesuai kebutuhan. Di dalam komponen, terdapat elemen TouchableOpacity yang memiliki dua properti utama:

* style={styles.container} untuk mengatur tampilan tombol.
* onPress={props.onPress} agar tombol menjalankan fungsi tertentu ketika ditekan.

Di dalam TouchableOpacity, terdapat elemen Text yang menampilkan tulisan dari props.text, yaitu teks yang dikirim dari komponen lain saat tombol digunakan.

Bagian akhir kode berisi pengaturan gaya menggunakan StyleSheet.create():

* container mengatur tampilan tombol dengan latar abu-abu muda (#dddddd), padding 15 piksel, posisi teks di tengah (alignItems: "center"), serta sudut membulat (borderRadius: 30).
* text mengatur ukuran huruf (fontSize: 16), membuat semua huruf kapital (textTransform: "uppercase"), dan menebalkan tulisan (fontWeight: "bold").

Terakhir, komponen Button diekspor dengan export default Button, sehingga dapat digunakan di file lain seperti App.js atau header.js.

# **Komponen Separator**

**Input:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Penjelasan:**

Baris pertama melakukan import terhadap elemen View dari React Native. Elemen View adalah wadah dasar yang biasa digunakan untuk menyusun komponen lain atau mengatur ruang kosong pada tampilan.

Selanjutnya dibuat komponen fungsional bernama Separator yang menerima parameter props. Komponen ini hanya menampilkan satu elemen View yang memiliki properti gaya (style) berisi height: props.height. Artinya, tinggi dari pemisah ini akan menyesuaikan dengan nilai yang dikirim melalui props saat komponen dipanggil.

Misalnya, jika Separator digunakan seperti <Separator height={30} />, maka komponen ini akan membuat jarak vertikal setinggi 30 piksel di antara dua elemen. Dengan begitu, pengembang tidak perlu menulis gaya jarak berulang kali, cukup memanggil komponen ini dengan nilai tinggi yang diinginkan.

Terakhir, Separator diekspor dengan export default Separator, sehingga bisa digunakan di file lain seperti App.js untuk memberikan jarak antar tombol atau komponen lainnya.

# **Screen List**

**Input:**

**A black screen with many lines

AI-generated content may be incorrect.**

**Penjelasan:**

Bagian pertama mendefinisikan data dummy berupa sebuah array berisi objek. Setiap objek memiliki tiga properti: id, title, dan image. Data ini berfungsi sebagai sumber informasi yang akan ditampilkan di layar. Misalnya, setiap item berisi judul berita dan tautan gambar dari situs resmi Telkom University Surabaya.

Setelah itu, dibuat komponen fungsional bernama List. Di dalamnya terdapat fungsi panah renderItem yang menerima parameter berupa objek item. Fungsi ini menentukan bagaimana setiap data ditampilkan di layar. Setiap item ditampilkan di dalam TouchableOpacity, yaitu elemen yang bisa ditekan oleh pengguna. Saat ditekan, akan muncul pesan pop-up menggunakan alert("Pressed").

Di dalam setiap TouchableOpacity, terdapat komponen Image untuk menampilkan gambar dari properti item.image, serta Text untuk menampilkan judul dari item.title.

Kemudian, komponen FlatList digunakan untuk menampilkan seluruh data dalam bentuk daftar yang bisa digulir (scrollable). Properti yang digunakan antara lain:

* data={datas} untuk menentukan sumber data yang akan ditampilkan.
* renderItem={renderItem} untuk memanggil fungsi yang menampilkan setiap item.
* keyExtractor={(item) => item.id} agar setiap item memiliki kunci unik berdasarkan ID-nya.

Selanjutnya, bagian StyleSheet digunakan untuk mengatur tampilan daftar agar terlihat rapi dan konsisten:

* view: memberikan padding 15 piksel serta garis pemisah di bagian bawah setiap item.
* image: menentukan tinggi gambar sebesar 200 piksel dan lebar mengikuti ukuran layar.
* text: mengatur ukuran huruf 18 dan memberi jarak 10 piksel dari gambar di atasnya.

Terakhir, komponen List diekspor menggunakan export default List, sehingga bisa digunakan di file App.js sebagai salah satu tampilan utama.

# **Screen Article**

**Input:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Penjelasan:**

Pertama, kode mengimpor beberapa elemen penting dari React Native seperti ScrollView, Image, View, Text, StyleSheet, dan Alert, serta dua komponen tambahan bernama Separator dan Button dari folder components.

Di dalam komponen Article, terdapat fungsi panah (arrow function) bernama buttonHandler yang berfungsi untuk menampilkan pesan peringatan (alert) bertuliskan “Button Handler” ketika tombol ditekan. Selanjutnya, bagian return menampilkan struktur tampilan utama. Semua konten dibungkus dengan elemen ScrollView, yang memungkinkan pengguna menggulir artikel panjang dari atas ke bawah.

Tampilan dimulai dengan logo Telkom University Surabaya di bagian atas, diikuti dengan judul artikel yang diletakkan di dalam View dengan latar belakang abu-abu muda. Setelah itu, isi artikel disusun dalam beberapa paragraf teks dan diselingi gambar serta pemisah antar bagian (Separator). Artikel ini berisi cerita tentang prestasi mahasiswa Telkom University Surabaya dalam ajang Innovillage 2023 dengan proyek berjudul Connect Pediatrics, yaitu sistem IoT dan website untuk memantau kesehatan balita di Posyandu.

Pada bagian bawah artikel, terdapat tombol Share yang jika ditekan akan menjalankan fungsi buttonHandler. Gaya tampilan atau pengaturan desain visual dibuat menggunakan StyleSheet.create, yang mengatur ukuran gambar, jarak antar elemen, warna latar, dan ukuran teks. Terakhir, komponen Article diekspor agar dapat digunakan di bagian lain dari aplikasi.

# **Output Keseluruhan**

**Output:**

**A group of people in a room

AI-generated content may be incorrect.**

**A screenshot of a cell phone

AI-generated content may be incorrect.**

# **Tugas**

**Input: list.js**

**A screen shot of a computer screen

AI-generated content may be incorrect.**

**Output:**

A screenshot of a cellphone

AI-generated content may be incorrect.

**Penjelasan:**

Kode yang telah diperbarui ini menambahkan fitur tampilan detail artikel menggunakan modal, sehingga pengguna bisa melihat informasi lebih lengkap ketika menekan salah satu item di daftar berita. Pada kode asli, daftar artikel hanya dapat ditekan dan menampilkan pesan sederhana berupa alert “Pressed”. Namun, dalam versi yang sudah diubah, ketika pengguna menekan salah satu artikel, akan muncul sebuah Modal interaktif yang berisi gambar, judul, dan deskripsi artikel yang dipilih.

Untuk mewujudkan hal tersebut, digunakan dua state baru dengan useState, yaitu modalVisible untuk mengatur apakah modal sedang ditampilkan atau tidak, dan selectedArticle untuk menyimpan data artikel yang sedang dipilih. Dua fungsi baru ditambahkan: openModal(item) untuk menampilkan modal sekaligus mengatur artikel mana yang dipilih, dan closeModal() untuk menutup modal serta menghapus data artikel dari state.

Selain itu, komponen Modal dan ScrollView ditambahkan agar isi detail artikel bisa digulir ke bawah jika kontennya panjang. Di dalam modal, digunakan komponen tambahan seperti Image, Text, Separator, dan Button agar tampilannya lebih menarik dan terstruktur. Tombol “Tutup” di bagian bawah modal memungkinkan pengguna menutup tampilan detail dengan mudah.

Bagian gaya tampilan (StyleSheet) juga diperluas dengan menambahkan pengaturan baru seperti modalOverlay untuk efek gelap di latar belakang ketika modal terbuka, modalContent untuk mengatur bentuk dan ukuran modal, serta modalImage, modalTitle, dan modalDescription untuk menyesuaikan tampilan gambar dan teks di dalam modal.

# **Link Repository GitHub:**