

Aplikasi Mobile Berita Online : Novea

Kelompok Aplikasi Novea TI.5A :

1. M Faiz Priyantoro (17223028)
2. Said Aqim Wjihatalillah (17223004)
3. Adhitya Rahadiansyah (17221014)
4. Lutfhi Muayyad Billah (17223011)
5. M Taufik Faturrohman (17223017)

Deskripsi Aplikasi Novea :

Novea berasal dari dua kata, "Nova" yang berarti baru, segar, dan penuh energi, serta "Era," yang mencerminkan masa kini dan masa depan yang terus berubah. Gabungan ini menunjukkan aplikasi berita yang menyajikan informasi terkini, relevan, dan inovatif yang mencerminkan perubahan dan perkembangan zaman. Kami berharap aplikasi Novea ini menjadi aplikasi yang responsif dan mudah digunakan untuk mengakses berita dari perangkat mobile.

Library yang Digunakan :

- **React Native:** Framework untuk membangun aplikasi mobile menggunakan JavaScript dan React.
- **React Navigation:** Untuk navigasi antar halaman dalam aplikasi.
- **React Native Paper:** Komponen UI yang mengikuti Material Design.
- **Lottie:** Untuk animasi yang menarik pada ikon navigasi.
- **News API:** Untuk mengambil data berita dari sumber berita online.

Struktur Halaman Aplikasi :

- **App.tsx :** Halaman utama yang mengatur navigasi dan tampilan dasar aplikasi.
- **Home.js :** Halaman yang menampilkan daftar berita dengan fitur pencarian.
- **News.js :** Halaman yang menampilkan detail berita yang dipilih.
- **Profile.js :** Halaman yang menampilkan deskripsi singkat aplikasi & anggota kelompok Novea.
- **Detail.js :** Halaman untuk menampilkan berita berdasarkan kategori tertentu.

Penjelasan Setiap Halaman :

1. App.tsx

Halaman utama yang berfungsi sebagai pengatur navigasi dan memberikan tampilan dasar untuk aplikasi. Di bawah ini, akan dijelaskan baris kode yang memiliki peran penting di dalam halaman ini.

```
1  const App = () => {  
2    const [index, setIndex] = useState(0);  
3    const [routes] = useState([  
4      { key: "home", title: "Home", icon: require("./screens/lottie/home.json") },  
5      { key: "profile", title: "Profile", icon: require("./screens/lottie/profile.json") },  
6      { key: "detail", title: "Detail", icon: require("./screens/lottie/detail.json") },  
7    ]);
```

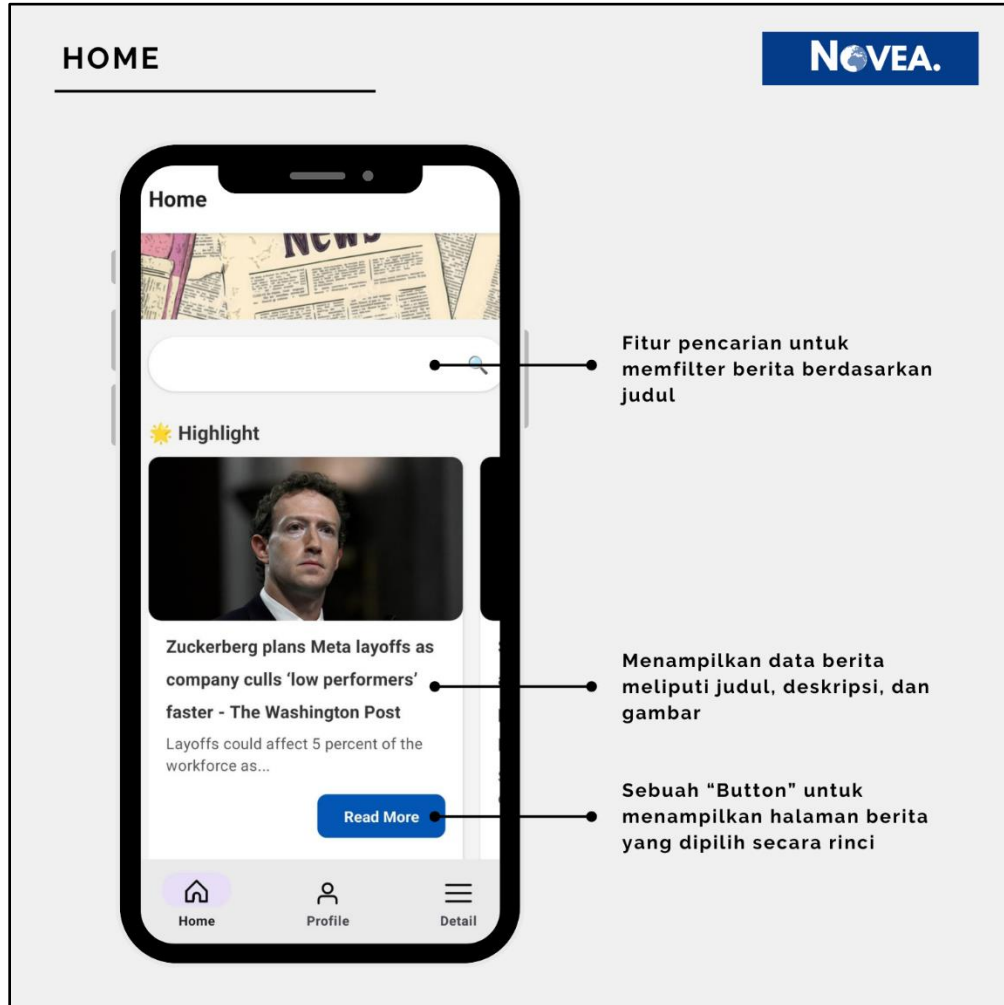
- **useState(0):** Menginisialisasi state index dengan nilai 0, yang menunjukkan halaman yang aktif yaitu halaman "Home".
- **useState([...]):** Menginisialisasi state routes yang berisi informasi tentang rute navigasi, termasuk kunci, judul, dan ikon Lottie untuk setiap halaman.

```
1  const renderScene = BottomNavigation.SceneMap({  
2    home: MemberHome,  
3    profile: Profile,  
4    detail: Detail,  
5  });
```

- **BottomNavigation.SceneMap:** Menghubungkan setiap kunci rute dengan komponen yang sesuai. Ini memungkinkan navigasi antar halaman dengan mudah.

Secara keseluruhan, dalam halaman **App.tsx** digunakan **BottomNavigation** sebagai metode navigasi antar halaman, yang mencakup halaman Home, Profile, dan Detail. Untuk mengelola halaman yang aktif, kami mengatur state yang memungkinkan pengguna untuk berpindah dengan mudah antara halaman-halaman tersebut. Selain itu, kami juga memanfaatkan **LottieView** untuk menampilkan animasi yang menarik pada ikon navigasi, sehingga meningkatkan pengalaman pengguna secara keseluruhan.

2. Home.js



Halaman 1. Home

Halaman ini telah dirancang untuk menampilkan daftar berita secara menyeluruh, lengkap dengan berbagai fitur pencarian yang dirancang untuk memudahkan pengguna dalam menemukan informasi yang mereka cari. Dengan adanya fitur pencarian ini, pengguna dapat dengan cepat dan efisien mencari berita yang relevan sesuai dengan kebutuhan mereka. Selanjutnya, akan dijelaskan juga setiap baris kode yang memiliki peran penting dalam halaman ini.

```

1  const fetchNews = async () => {
2    setLoading(true);
3    setError(null);
4    try {
5      const apiKey = "46b109782549481697c78b3a047cebc4";
6      const response = await fetch(`https://newsapi.org/v2/top-headlines?
country=us&apiKey=${apiKey}`);
7      const data = await response.json();

```

- **fetchNews:** Fungsi asinkron yang mengambil data berita dari News API.
- **setLoading(true):** Mengatur state loading menjadi true untuk menunjukkan bahwa data sedang diambil.
- **await fetch(...):** Mengambil data dari API menggunakan fetch dan menunggu respons.

```

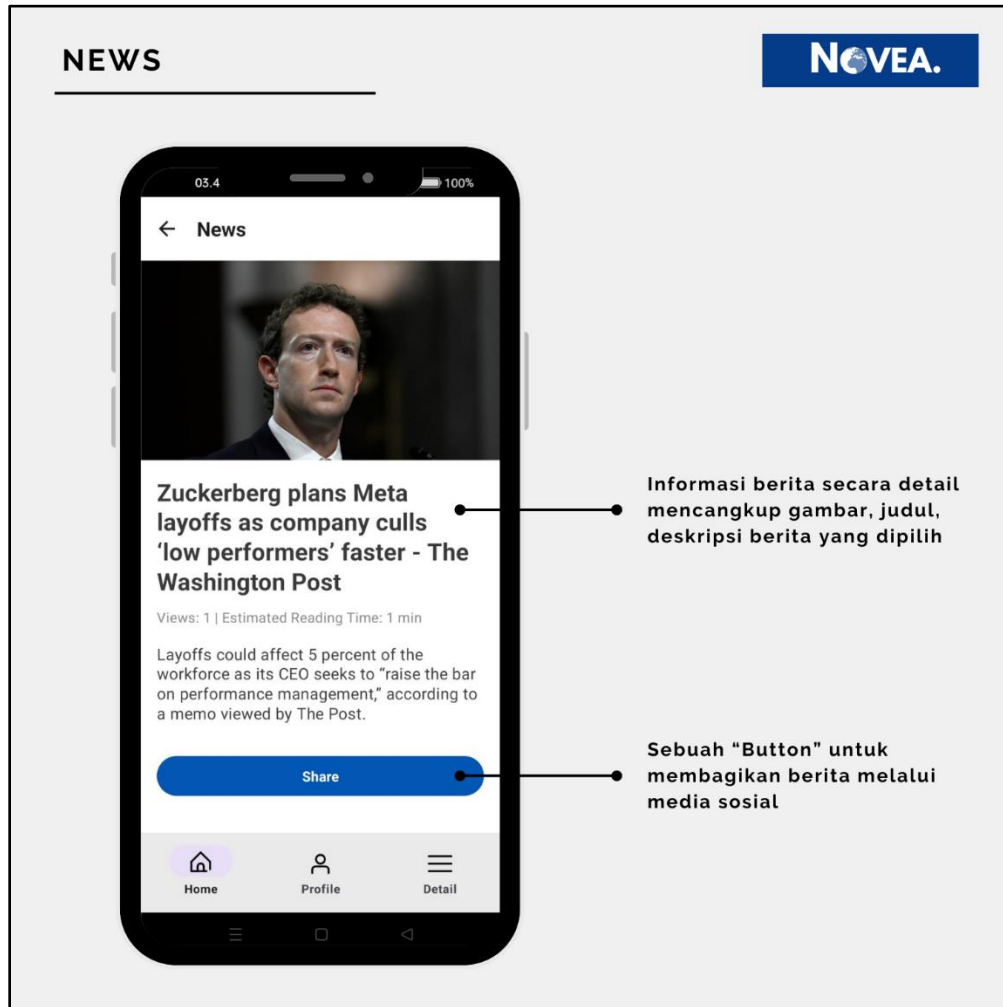
1  if (data.articles) {
2    const formattedData = data.articles
3      .filter(article => article.title && article.urlToImage)
4      .map((article, index) => ({
5        id: index.toString(),
6        title: article.title,
7        description: article.description || "No Description",
8        image: article.urlToImage || "https://via.placeholder.com/150",
9      }));

```

- Memeriksa apakah ada artikel dalam data yang diterima. Jika ada, artikel difilter untuk memastikan hanya artikel dengan judul dan gambar yang valid yang akan ditampilkan.

Secara keseluruhan, halaman **Home.js** mengambil data berita dari News API dengan menggunakan metode fetch. Data berita yang berhasil diambil kemudian disimpan dalam state dan ditampilkan dalam format kartu untuk memudahkan pengguna dalam membaca informasi. Kami juga menyediakan fitur pencarian yang memungkinkan pengguna untuk memfilter berita berdasarkan judul, sehingga mereka dapat dengan cepat menemukan informasi yang relevan.

3. News.js



Halaman 2. News

Halaman ini memiliki fungsi utama untuk menampilkan rincian berita yang telah dipilih oleh pengguna, sehingga mereka dapat memperoleh informasi yang lebih mendalam mengenai topik yang diminati. Di bagian berikut, akan dijelaskan secara mendetail setiap baris kode yang memiliki kontribusi penting dalam halaman ini.

```
1 const { title, description, image, url } = route.params;
```

- Mengambil parameter dari rute yang diterima, yang berisi informasi tentang berita yang dipilih (judul, deskripsi, dan gambar).

```
1 const shareNews = async () => {  
2   try {  
3     await Share.share({  
4       message: `${title}\n\n${url}`,  
5     });  
6   } catch (error) {  
7     console.log(error.message);  
8   }  
9 };
```

- **shareNews:** Untuk membagikan berita melalui media sosial. Menggunakan “Share.share” untuk membuka dialog berbagi dengan judul dan URL berita.

Selanjutnya, halaman **News.js** menampilkan detail berita yang dipilih dari halaman Home, memberikan tampilan yang lebih mendalam terhadap berita yang dipilih. Untuk meningkatkan pengalaman membaca, kami juga menghitung jumlah tampilan dan estimasi waktu baca berdasarkan jumlah kata dalam deskripsi berita, sehingga pengguna dapat memperkirakan berapa lama mereka akan menghabiskan waktu untuk membaca. Selain itu, kami menyediakan fitur untuk membagikan berita melalui media sosial, memungkinkan pengguna untuk dengan mudah menyebarkan informasi yang mereka anggap menarik kepada teman-teman mereka.

4. Profile.js

- **Data Novea dan Tim**

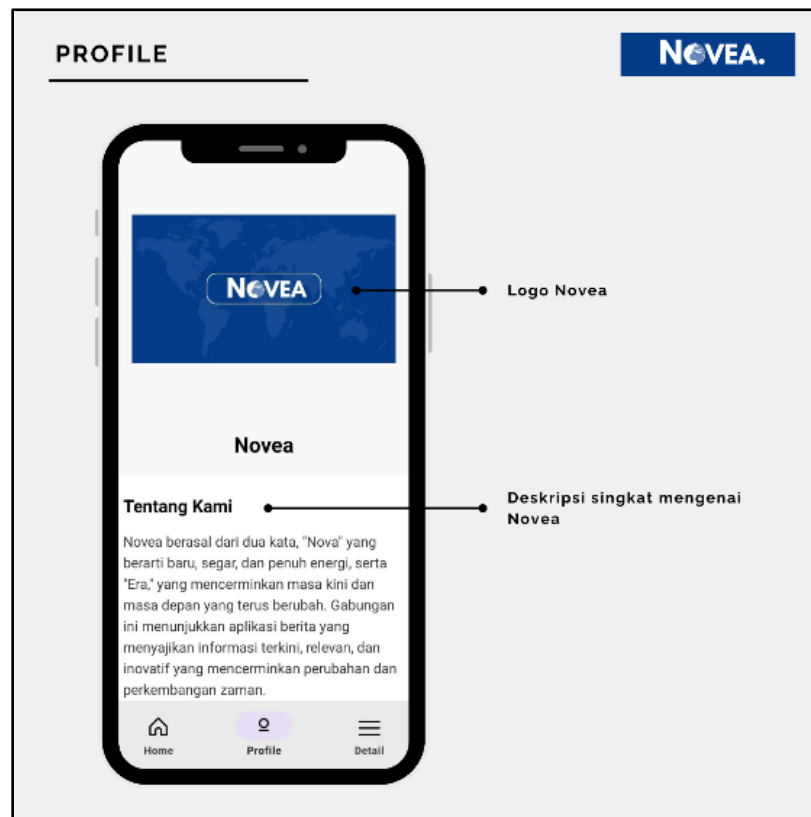
Kode ini dimulai dengan mendefinisikan data perusahaan dan anggota tim menggunakan objek JavaScript. Contohnya:

```
1 const companyData = {  
2   name: 'Novea',  
3   logo: 'novea.jpg',  
4   description: 'Novea berasal dari dua kata, "Nova" yang berarti baru...',  
5   address: 'Jl. Novea No. 123, Bandung',  
6   phone: '+62 96 7500 6525',  
7   email: 'novea@trusted.id',  
8   Instagram: 'novea.id',  
9 };
```

Data ini digunakan untuk menampilkan nama perusahaan, logo, deskripsi, kontak, dan akun Instagram.

- **Tampilan Utama Halaman**

Berikutnya, kita menggunakan komponen React Native seperti ``, ``, ``, dan `` untuk menyusun tata letak halaman. Bagian ini menampilkan header Novea dan deskripsi:



Halaman 3. Profile

```

1 <ScrollView style={styles.container}>
2   <View style={styles.headerContainer}>
3     <Image
4       source={images[companyData.logo]}
5       style={styles.logo}
6       resizeMode="contain"
7     />
8     <Text style={styles.companyName}>{companyData.name}</Text>
9   </View>
10
11  <View style={styles.sectionContainer}>
12    <Text style={styles.sectionTitle}>Tentang Kami</Text>
13    <Text style={styles.description}>{companyData.description}</Text>
14  </View>
15
16  <View style={styles.sectionContainer}>
17    <Text style={styles.sectionTitle}>Kontak</Text>
18    <Text style={styles.contactInfo}>Alamat: {companyData.address}</Text>
19    <Text style={styles.contactInfo}>Telepon: {companyData.phone}</Text>
20    <Text style={styles.contactInfo}>Email: {companyData.email}</Text>
21    <Text style={styles.contactInfo}>Instagram: {companyData.instagram}</Text>
22  </View>
23 </ScrollView>

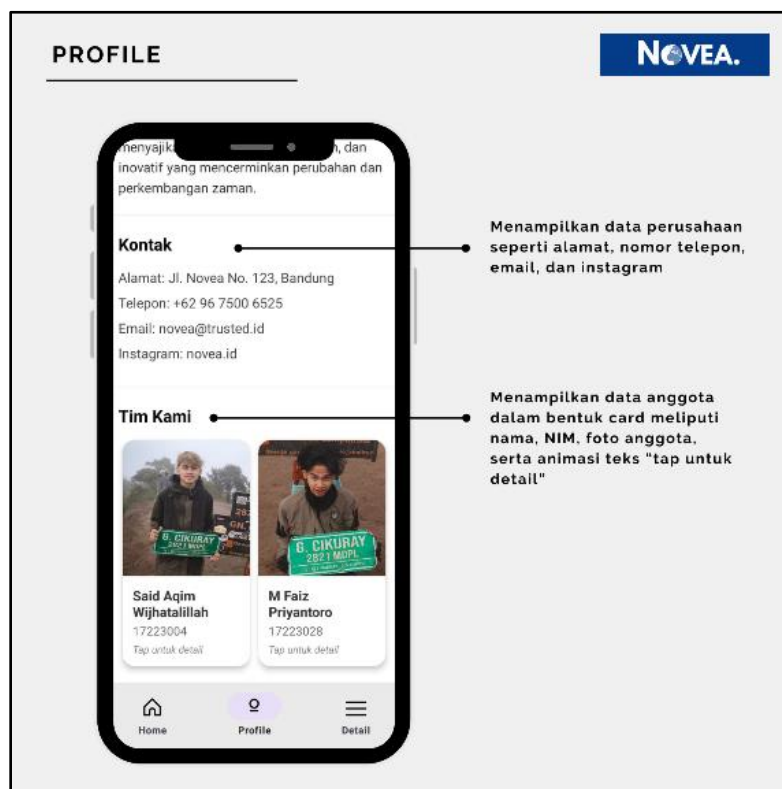
```

<ScrollView> memungkinkan pengguna menggulir halaman jika kontennya panjang.

<Image> digunakan untuk menampilkan logo Novea, sementara **<Text>** untuk deskripsi, dan informasi lainnya.

- **Kartu Profil Anggota Tim**

Bagian ini menampilkan daftar anggota tim meliputi nama, posisi, dan gambar anggota:




```

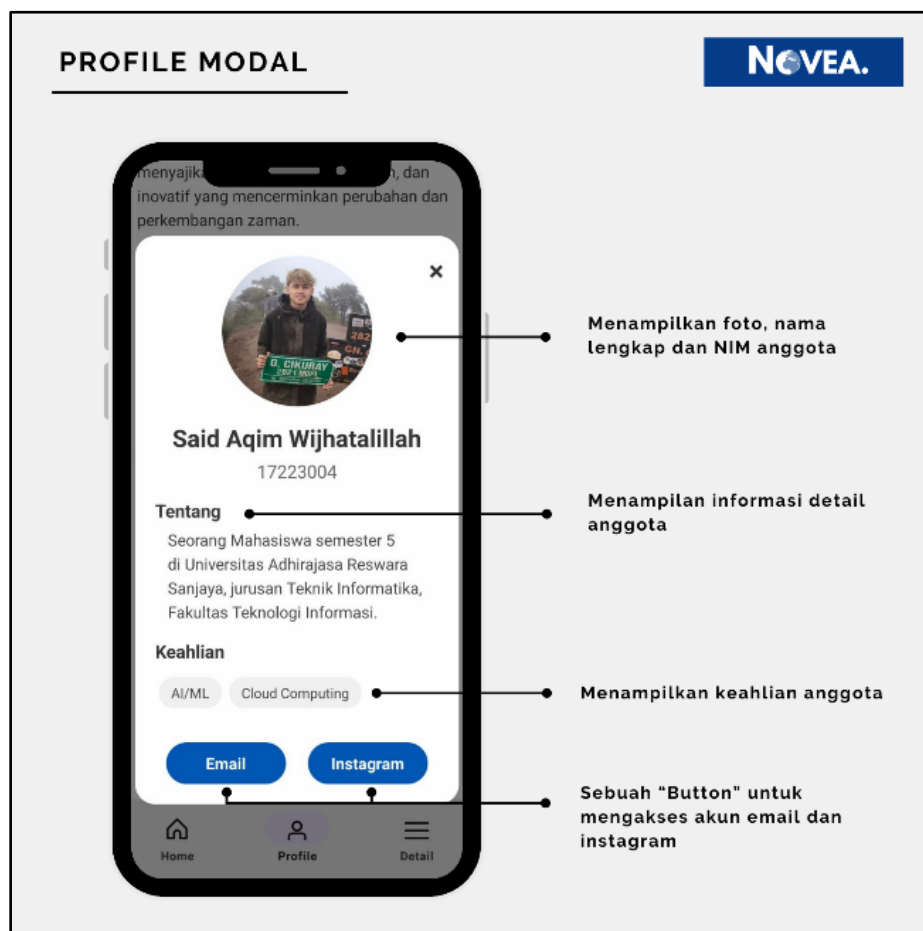
1 <View style={styles.teamContainer}>
2   {teamMembers.map(member => (
3     <TeamMemberCard key={member.id} member={member} />
4   ))}
5 </View>

```

TeamMemberCard adalah komponen khusus yang menangani logika tampilan dan interaksi setiap anggota tim. Ketika kartu ditekan, modal popup akan muncul menampilkan detail anggota.

- **Modal PopUp Interaktif**

Modal digunakan untuk menampilkan informasi detail anggota tim, seperti bio dan keahlian mereka. Inilah cuplikan tampilan dan kode untuk modal:



```

1  <Modal
2    animationType="slide"
3    transparent={true}
4    visible={modalVisible}
5    onRequestClose={onClose}>
6    <View style={styles.modalOverlay}>
7      <View style={styles.modalContent}>
8        <TouchableOpacity style={styles.closeButton} onPress={onClose}>
9          <Text style={styles.closeButtonText}>x</Text>
10       </TouchableOpacity>
11
12       <Image
13         source={images[member.image]}
14         style={styles.modalImage}
15         resizeMode="cover"
16       />
17
18       <Text style={styles.modalName}>{member.name}</Text>
19       <Text style={styles.modalPosition}>{member.position}</Text>
20       <Text style={styles.modalBio}>{member.bio}</Text>
21     </View>
22   </View>
23 </Modal>

```

Bagaimana Modal Ini Bekerja?

`Modal` adalah komponen bawaan React Native untuk membuat popup. Properti seperti `visible` mengontrol apakah modal terlihat atau tidak. Ketika kartu ditekan, state `modalVisible` diubah menjadi `true`, sehingga modal muncul seperti gambar, nama, dan bio anggota tim diambil dari data anggota.

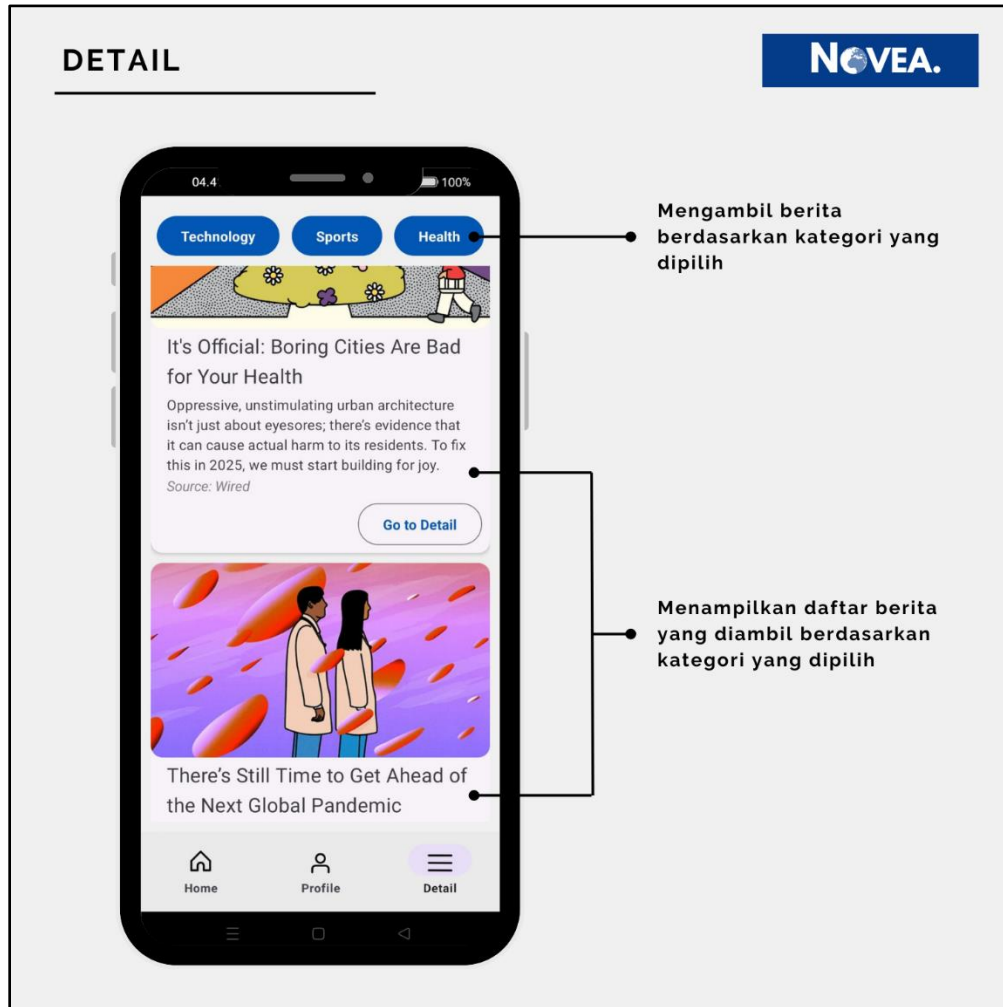
- **Animasi Kartu**

Bagian ini menambahkan sedikit efek animasi ketika pengguna menekan kartu anggota tim. Dengan menggunakan `Animated`, kita bisa membuat efek "zoomin" yang halus:

```
1  const handlePressIn = () => {
2    Animated.spring(scaleAnim, {
3      toValue: 0.95,
4      useNativeDriver: true,
5    }).start();
6  };
7
8  const handlePressOut = () => {
9    Animated.spring(scaleAnim, {
10     toValue: 1,
11     useNativeDriver: true,
12   }).start();
13  };
```

Secara keseluruhan, halaman **Profile.js** menampilkan informasi mengenai aplikasi, termasuk nama aplikasi, deskripsi singkat, informasi kontak yang dapat dihubungi, serta logo aplikasi yang mencerminkan identitas kami. Selain itu, kami juga menyediakan daftar anggota tim yang dilengkapi dengan kartu profil masing-masing, yang menampilkan foto dan nama mereka dalam tim. Untuk memberikan pengalaman yang lebih interaktif, kami telah menambahkan fitur modal popup yang memungkinkan pengguna untuk melihat detail lebih lanjut tentang setiap anggota tim ketika mereka mengklik kartu profil tersebut. Dengan demikian, pengunjung dapat dengan mudah mengenal lebih dekat dengan anggota yang berkontribusi dalam pengembangan aplikasi ini.

- Detail.js



Halaman 4. Detail

Halaman ini dibuat untuk menampilkan berita yang dikelompokkan berdasarkan kategori tertentu, sehingga pengguna dapat dengan mudah menemukan informasi yang relevan sesuai dengan minat atau kebutuhan mereka. Dengan adanya pengelompokan berdasarkan kategori, pengguna dapat menjelajahi berbagai topik dengan lebih terfokus dan efisien. Selanjutnya, akan dijelaskan juga mengenai setiap baris kode yang memiliki kontribusi penting dalam halaman ini.

```

1  const fetchNewsByCategory = async (category) => {
2    const url = `https://newsapi.org/v2/everything?
    q=${category}&apiKey=${apiKey}`;
3    try {
4      setLoading(true);
5      const response = await fetch(url);

```

- **fetchNewsByCategory:** Fungsi yang mengambil berita berdasarkan kategori yang dipilih.
- **setLoading(true):** Mengatur state loading menjadi true untuk menunjukkan bahwa data sedang diambil.

```

1  <FlatList
2    data={articles}
3    keyExtractor={(item, index) => index.toString()}
4    renderItem={({ item }) => (
5      <Card style={styles.articleContainer}>
6        {item.urlToImage && (
7          <Card.Cover source={{ uri: item.urlToImage }}
8        />
9      )}

```

- **FlatList:** Komponen untuk menampilkan daftar berita berdasarkan kategori yang dipilih

Secara keseluruhan, halaman **Detail.js** memungkinkan pengguna untuk mengambil berita berdasarkan kategori yang dipilih, seperti teknologi, olahraga, dan kesehatan. Setelah kategori ditentukan, kami menampilkan daftar artikel yang relevan sesuai dengan pilihan tersebut. Untuk menampilkan daftar artikel dengan efisien, kami menggunakan komponen FlatList, yang memungkinkan pengelolaan data yang baik dan meningkatkan performa aplikasi saat menampilkan banyak artikel sekaligus.

Kesimpulan

Aplikasi berita online yang dibangun menggunakan React Native ini menawarkan pengalaman pengguna yang intuitif dengan fitur bottom-navigation yang memudahkan perpindahan antar halaman. Pengguna dapat dengan cepat mencari berita berdasarkan judul, melihat detail berita yang mencakup deskripsi dan gambar, serta membagikannya melalui media sosial. Selain itu, aplikasi ini menyediakan kategorisasi berita, memungkinkan pengguna untuk menjelajahi artikel berdasarkan minat mereka, seperti teknologi, olahraga, dan kesehatan. Secara keseluruhan, aplikasi ini memberikan akses yang efisien dan terstruktur terhadap berita terkini.