




React Native Animasi





Apa itu Animasi?

Animasi adalah teknik membuat objek bergerak atau berubah seiring waktu. Dalam pengembangan aplikasi mobile, animasi dapat meningkatkan pengalaman pengguna dan membuat aplikasi terlihat lebih menarik.



Mengapa Animasi Penting?

1. Peningkatan Pengalaman Pengguna (UX): Animasi membuat antarmuka pengguna lebih responsif dan menyenangkan.
2. Menonjolkan Perubahan Keadaan: Animasi membantu menyoroti perubahan keadaan atau transisi antar elemen.
3. Menyampaikan Informasi Lebih Baik: Dengan animasi, informasi dapat disampaikan dengan lebih efektif dan intuitif.




Jenis Animasi di React Native

1. Built-in Animations

- *Animated* API

2. Third-party Libraries

- React-native-reanimated
 - React-native-animatable
- 

Animated API

- Core Component:
 - *Animated.Value*: Nilai yang dapat dianimasikan
 - *Animated.timing*: Animasi dengan durasi tertentu
 - *Animated.spring*: Animasi dengan efek pegas
 - *Animated.sequence* dan *Animated.parallel*: Menggabungkan beberapa animasi

Contoh Simple Animation

Menggerakkan Elemen

```
1 import React, { useRef } from 'react';
2 import { Animated, View, Button, StyleSheet } from 'react-native';
3
4 const App = () => {
5   const position = useRef(new Animated.Value(0)).current;
6
7   const move = () => {
8     Animated.timing(position, {
9       toValue: 100,
10      duration: 1000,
11      useNativeDriver: true,
12    }).start();
13  };
14
15  return (
16    <View style={styles.container}>
17      <Animated.View style={[styles.box, { transform: [{ translateY: position }] }]} />
18      <Button title="Move" onPress={move} />
19    </View>
20  );
21 };
22
23 const styles = StyleSheet.create({
24   container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
25   box: { width: 50, height: 50, backgroundColor: 'blue' },
26 });
27
28 export default App;
29
```

Advanced Animations

- `Animated.sequence`
 - Jalankan beberapa animasi secara berurutan



```
1  Animated.sequence([  
2    Animated.timing(value1, { toValue: 100, duration: 500 }),  
3    Animated.timing(value2, { toValue: 200, duration: 500 }),  
4  ]).start();  
5
```

- `Animated.parallel`
 - Jalankan beberapa animasi secara bersamaan.

Libraries Pihak Ketiga

1. react-native-reanimated
 - Lebih powerful dan modern.
 - Cocok untuk animasi kompleks
2. react-native-animatable
 - Animasi sederhana dengan code minimal

```
1  import * as Animatable from 'react-native-animatable';
2
3  const App = () => (
4    <Animatable.Text animation="zoomIn" duration={1000}>
5      Hello, Animations!
6    </Animatable.Text>
7  );
8
```


Praktik Terbaik

- Gunakan useNativeDriver untuk performa lebih baik
- Sesuaikan animasi dengan UX aplikasi
- Hindari penggunaan animasi berlebih yang dapat mengganggu pengguna

Kesimpulan

- React Native menyediakan banyak opsi untuk animasi.
- Gunakan animasi untuk memperkaya pengalaman pengguna.
- Pelajari juga library pihak ketiga untuk kebutuhan lebih kompleks.



Thank
You

