

Firestore

Raih Mimpi #TanpaBatas



Manage multiple permissions

- ☐ Pengenalan
- ☐ Config Firebase
- ☐ CRUD Operations
- ☐ Membuat aplikasi catatan sederhana

Objektif sesi

- Pemahaman Integrasi React Native dan Firestore
- Pembelajaran Dasar CRUD dalam Firestore
- Membuat aplikasi catatan sederhana

Pengenalan

Firestore adalah database NoSQL yang fleksibel dan skalabel dari Firebase, yang sangat cocok untuk pengembangan aplikasi mobile dengan React Native. Integrasi ini memungkinkan Anda menyimpan dan sinkronisasi data antar pengguna dalam waktu nyata dengan mudah.

Config Firestore

- Instalasi

```
npm install --save firebase
```

Atau

```
yarn add firebase
```

Config Firestore

- Config firebase

```
1 import * as firebase from 'firebase';
2 import firestore from 'firebase/firestore'
3
4 const firebaseConfig = {
5   apiKey: "",
6   authDomain: "",
7   databaseURL: "",
8   projectId: "",
9   storageBucket: "",
10  messagingSenderId: "",
11  appId: "",
12  measurementId: ""
13 };
14
15 firebase.initializeApp(firebaseConfig);
16
17 firebase.firestore();
18
19 export default firebase;
```

Config Firestore

- **Menggunakan Firestore**

Untuk menggunakan Firestore, import database dari instance Firebase

```
import firebase from './firebaseConfig';  
  
const db = firebase.firestore();
```

CRUD Operations

- Membaca Data

```
db.collection("nama_koleksi")  
  .get()  
  .then(querySnapshot => {  
    querySnapshot.forEach(documentSnapshot => {  
      console.log(documentSnapshot.id, documentSnapshot.data());  
    });  
  });
```


CRUD Operations

- Menambah Data

```
db.collection("nama_koleksi").add({  
  field1: "value1",  
  field2: "value2",  
  // lebih banyak fields...  
})  
  
.then(docRef => {  
  console.log("Document written with ID: ", docRef.id);  
})  
  
.catch(error => {  
  console.error("Error adding document: ", error);  
});
```

CRUD Operations

- **Memperbarui Data**

```
db.collection("nama_koleksi").doc("document_id")  
  .update({  
    field1: "new_value"  
    // lebih banyak updates...  
  });
```

CRUD Operations

- Menghapus Data

```
db.collection("nama_koleksi").doc("document_id").delete();
```

CRUD Operations

- Realtime Data

```
db.collection("nama_koleksi")
  .onSnapshot(querySnapshot => {
    querySnapshot.docChanges().forEach(change => {
      if (change.type === "added") {
        console.log("New data: ", change.doc.data());
      }
      // Handle changes...
    });
  });
```

Membuat aplikasi catatan sederhana

- **Struktur folder**

```
MyNoteApp/  
├─ src/  
│   ├─ components/  
│   │   └─ NoteItem.js  
│   ├─ screens/  
│   │   └─ HomeScreen.js  
│   └─ config/  
│       └─ firebaseConfig.js  
├─ App.js  
└─ package.json
```


Membuat aplikasi catatan sederhana

- **Keterangan**

- src/: Folder utama untuk kode sumber aplikasi.
- components/: Berisi komponen-komponen yang digunakan dalam aplikasi.
- screens/: Berisi layar atau halaman aplikasi.
- config/: Berisi konfigurasi, seperti pengaturan Firebase.
- App.js: File utama yang mengatur navigasi dan struktur aplikasi.
- package.json: Berisi metadata dan dependensi proyek.

Membuat aplikasi catatan sederhana

- **Setup Firestore Database**

Dalam Firebase Console, di sidebar, klik "Firestore Database".

Klik "Create database" dan pilih mode yang diinginkan (test mode atau production mode).

Atur lokasi Cloud Firestore sesuai kebutuhan.

- **Membuat Collection**

Dalam Firestore Database, buatlah sebuah collection (misalnya dengan nama "notes") yang akan digunakan untuk menyimpan catatan.

Anda dapat menambahkan dokumen secara manual melalui Firebase Console atau melalui aplikasi Anda.

Membuat aplikasi catatan sederhana

- **HomeScreen**

```
1 import React, { useState, useEffect } from 'react';
2 import { View, Text, TextInput, Button, FlatList } from 'react-native';
3 import db from '../config/firebaseConfig';
4
5 const HomeScreen = () => {
6   const [note, setNote] = useState('');
7   const [notes, setNotes] = useState([]);
8
9   useEffect(() => {
10     const unsubscribe = db.collection('notes').onSnapshot(snapshot => {
11       const fetchedNotes = snapshot.docs.map(doc => ({
12         id: doc.id,
13         ...doc.data()
14       }));
15       setNotes(fetchedNotes);
16     });
17     return () => unsubscribe();
18   }, []);
19
20   const addNote = () => {
21     db.collection('notes').add({ text: note });
22     setNote('');
23   };
24
25   const deleteNote = id => {
26     db.collection('notes').doc(id).delete();
27   };
28
29   return (
30     <View>
31       <TextInput
32         placeholder="Enter note"
33         value={note}
34         onChangeText={setNote}
35       />
36       <Button title="Add Note" onPress={addNote} />
37       <FlatList
38         data={notes}
39         renderItem={({ item }) => (
40           <Text onPress={() => deleteNote(item.id)}>{item.text}</Text>
41         )}
42         keyExtractor={item => item.id}
43       />
44     </View>
45   );
46 };
47
48 export default HomeScreen;
49
50
```



Thank you