

Firebase Authentication Setup

Langkah-langkah Setup Firebase

1. Buat Firebase Project

1. Buka [Firebase Console](#)
2. Klik "Add project" atau "Tambah project"
3. Masukkan nama project (misal: ai-customerservice)
4. Ikuti langkah-langkah setup

2. Enable Authentication

1. Di Firebase Console, pilih project Anda
2. Klik "Authentication" di menu sebelah kiri
3. Klik "Get started"
4. Pilih tab "Sign-in method"
5. Enable "Email/Password"

3. Buat Web App

1. Di Firebase Console, klik icon gear (⚙️) > Project settings
2. Scroll ke bawah ke "Your apps"
3. Klik icon web (</>)
4. Daftar app dengan nickname (misal: ai-customerservice-web)
5. Copy configuration yang diberikan

4. Setup Firestore (Optional untuk menyimpan data user tambahan)

1. Di Firebase Console, klik "Firestore Database"
2. Klik "Create database"
3. Pilih mode "Production" atau "Test"
4. Pilih location (misal: asia-southeast2 untuk Jakarta)

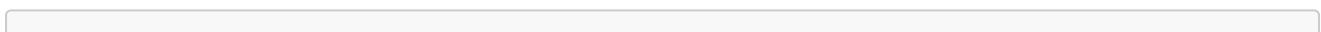
5. Generate Service Account Key (untuk Admin SDK)

1. Di Firebase Console, klik icon gear (⚙️) > Project settings
2. Klik tab "Service accounts"
3. Klik "Generate new private key"
4. Simpan file JSON yang di-download

6. Setup Environment Variables

Buka file `.env.local` dan isi dengan credentials dari Firebase:

Dari Service Account Key (file JSON yang di-download):



```
FIREBASE_PROJECT_ID=your-project-id
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----
\nYOUR_PRIVATE_KEY_HERE\n-----END PRIVATE KEY-----\n"
FIREBASE_CLIENT_EMAIL.firebaseio-adminsdk-xxxxxx@your-project-
id.iam.gserviceaccount.com
```

Catatan: Untuk FIREBASE_PRIVATE_KEY, copy nilai dari field "private_key" di file JSON dan pastikan semua \n tetap ada.

Dari Firebase Web App Config:

```
NEXT_PUBLIC_FIREBASE_API_KEY=AIzaSyXXXXXXXXXXXXXXXXXXXX
NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=your-project-id.firebaseio.com
NEXT_PUBLIC_FIREBASE_PROJECT_ID=your-project-id
NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET=your-project-id.appspot.com
NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID=123456789012
NEXT_PUBLIC_FIREBASE_APP_ID=1:123456789012:web:abcdef123456
```

7. Install Dependencies (sudah dilakukan)

```
npm install firebase firebase-admin express dotenv cors
```

8. Jalankan Development Server

```
npm run dev
```

PROF

Struktur File Authentication

```
src/
  app/
    api/
      auth/
        register/route.js    # API endpoint untuk registrasi
        login/route.js       # API endpoint untuk login
        verify/route.js      # API endpoint untuk verify token
      (public)/
        login/page.js        # Halaman login
        register/page.js     # Halaman registrasi
        layout.js             # Root layout dengan AuthProvider
    components/
      auth/
        LoginForm.js         # Form login
```

```
|       └── RegisterForm.js      # Form registrasi
|           └── ProtectedRoute.js # Component untuk protected routes
|   └── contexts/
|       └── AuthContext.js      # Auth context untuk state
management
└── lib/
    └── firebase.js            # Firebase client config
        └── firebaseAdmin.js     # Firebase admin config
```

Cara Menggunakan

1. Registrasi User Baru

- Akses [/register](#)
- Isi form dengan nama, email, dan password
- User otomatis login setelah registrasi berhasil

2. Login

- Akses [/login](#)
- Isi email dan password
- Redirect ke halaman customer setelah berhasil

3. Protect Routes

Untuk melindungi halaman yang memerlukan authentication:

```
— PROF —
import ProtectedRoute from '@/components/auth/ProtectedRoute';

export default function CustomerPage() {
  return (
    <ProtectedRoute>
      {/* Your page content here */}
    </ProtectedRoute>
  );
}
```

4. Menggunakan Auth Context

```
'use client';
import { useAuth } from '@/contexts/AuthContext';

export default function MyComponent() {
  const { user, login, logout, register } = useAuth();

  // user berisi: { uid, email, displayName, idToken }
  // atau null jika belum login
```

```
return (
  <div>
    {user ? (
      <div>
        <p>Welcome, {user.displayName}</p>
        <button onClick={logout}>Logout</button>
      </div>
    ) : (
      <p>Please login</p>
    )}
  </div>
);
}
```

API Endpoints

POST /api/auth/register

Registrasi user baru

Request:

```
{
  "email": "user@example.com",
  "password": "password123",
  "name": "John Doe"
}
```

Response:

```
{
  "message": "User berhasil didaftarkan",
  "user": {
    "uid": "...",
    "email": "user@example.com",
    "name": "John Doe"
  },
  "customToken": "..."
}
```

POST /api/auth/login

Login user

Request:

```
{
  "idToken": "firebase-id-token"
}
```

Response:

```
{  
  "message": "Login berhasil",  
  "user": {  
    "uid": "...",  
    "email": "user@example.com",  
    "name": "John Doe"  
  }  
}
```

POST /api/auth/verify

Verifikasi token

Request:

```
{  
  "idToken": "firebase-id-token"  
}
```

Response:

```
{  
  "valid": true,  
  "user": {  
    "uid": "...",  
    "email": "user@example.com",  
    "name": "John Doe"  
  }  
}
```

Troubleshooting

Error: "app/invalid-credential" atau "auth/invalid-credential"

PROF

- Pastikan service account key sudah benar
- Cek apakah FIREBASE_PRIVATE_KEY sudah benar formatnya dengan semua \n

Error: "auth/email-already-exists"

- Email sudah terdaftar, gunakan email lain atau login

Error: "auth/wrong-password" atau "auth/user-not-found"

- Email atau password salah saat login

Error: "Firebase Admin not initialized"

- Pastikan environment variables sudah benar
- Restart development server

Security Notes

1. Jangan commit file `.env.local` ke git
2. Jangan share service account key ke public
3. Enable App Check untuk production
4. Setup Firebase Security Rules untuk Firestore
5. Gunakan HTTPS di production