**LAPORAN PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK**



**JUDUL:**

**Authentikasi Firebase**

Disusun oleh:

Faiq Mufrih Hakim (21102245)

**TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

**Pembahasan Firebase**

Firebase Authentication adalah layanan yang disediakan oleh Google Firebase untuk mengelola otentikasi pengguna di aplikasi Anda. Dengan menggunakan Firebase Authentication, pengembang dapat dengan mudah menambahkan fitur otentikasi menggunakan email dan kata sandi, nomor telepon, serta metode otentikasi berbasis identitas pihak ketiga seperti Google, Facebook, dan Twitter. Firebase menyediakan SDK yang memungkinkan integrasi otentikasi menjadi lebih sederhana dan konsisten di berbagai platform seperti web, iOS, dan Android. Hal ini mengurangi beban pengembang dalam mengelola infrastruktur otentikasi dan keamanan pengguna secara manual.

Pada inti operasinya, Firebase Authentication menggunakan token ID berbasis JSON Web Token (JWT) untuk mengidentifikasi pengguna secara aman. Setelah pengguna berhasil masuk, Firebase Authentication mengeluarkan token ID yang mencakup informasi identitas pengguna. Token ini kemudian dapat digunakan untuk mengautentikasi permintaan yang dibuat oleh aplikasi ke server backend. Firebase Authentication juga menyediakan kemampuan untuk mengelola pengguna seperti membuat pengguna baru, menghapus pengguna, memperbarui informasi pengguna, dan mengelola kata sandi.

Keunggulan utama dari Firebase Authentication adalah kemudahan integrasi dan keamanan yang disediakan. Firebase menangani banyak aspek kompleks dari otentikasi pengguna seperti penyimpanan data pengguna dengan aman, perlindungan terhadap serangan keamanan seperti brute force, dan pemulihan akun pengguna yang terlupa kata sandi. Selain itu, Firebase Authentication terintegrasi dengan layanan Firebase lainnya seperti Firestore dan Realtime Database, memungkinkan pengembang untuk dengan mudah membangun aplikasi yang aman dan skalabel dengan otentikasi pengguna yang solid.

**Langkah-langkah Praktikum**

|  |  |
| --- | --- |
| **Langkah Praktikum** | **Pembahasan** |
|  | Membuat akun di Firebase website |
|  | Mengunduh firebase CLI dan NodeJs |
|  | *Install* node.js |
|  | Instal Firebase CLI melalui npm dengan menjalankan perintah berikut:  **npm install -g firebase-tools** |
|  | Membuat Project baru di firebase. |
|  | Menambahkan *flutter* ke *firebase.* |
|  | Langkah pertama dalam menambahkan *flutter* ke *firebase* adalah dengan masuk ke *firebase* lewat *command prompt* dengan mengetikan perintah “*firebase*  *login”.* |
|  | Setelah login ke *firebase* melalui *command prompt,* jika baru pertama kali melakukan, maka akan diarahkan untuk masuk ke akun yang terhubung dengan  *firebase CLI.* |
|  | Setelah melakukan tahapan-tahapan sebelumnya, halaman sukses akan muncul seperti gambar di samping. |
|  | Membuat project flutter baru. |
| *l* | menjalankan perintah “dart pub global activate flutterfire\_cli” membuat tool flutterfire dapat diakses dari mana saja di terminal |
|  | Perintah flutterfire configure -- project=prakflutterfirebase-38bb6 digunakan untuk mengkonfigurasikan firebase ke dalam project flutter. |
|  | Project berhasil dikonfigurasi. Dan berhasil mendapatkan firebase AppId |
|  | Menginstall dependensi dengan menggunakan perintah Ketik “flutter pub add firebase\_core” |
|  | Mengubah void main |
|  | Membuat kode untuk mengimplementasikan desain antarmuka pengguna pada gambar di samping. |
|  | Membuat folder baru yang diberi nama “ui”, dan di dalam folder tersebut buatlah 4 file dart sesuai pada gambar di samping. |
|  | Tambahkan kode di samping pada file splash.dart |
|  | Tambahkan kode di samping pada bagian file login.dart |
|  | Tambahkan kode di samping pada bagian file register.dart |
|  | Tambahkan kode di samping pada bagian home\_screen.dart |
|  | Selanjutnya kita perlu mendaftarkan state yang sudah kita buat dalam route dengan cara buat folder “utils” di dalam folder “lib”, dan buat file “routes.dart” di dalam folder”  utils”. |
|  | Tambahkan kode di samping pada bagian file routes.dart |
|  | Tambahkan kode di samping pada bagian file main.dart |
|  | Menambahkan authentikasi email dan password pada dashboard firebase di project kita. |
|  | Menginstal dependensi firebase\_auth |
|  | Cek ketersediaan dependensi firebase\_auth pada pubspec.yaml |
|  | Untuk memproses logika bisnis pada aplikasi kita akan menggunakan flutter bloc,  sehingga perlu menambahkan 2 dependensi yaitu :  bloc: ^8.1.2 flutter\_bloc: ^8.1.3  jangan lupa untuk menjalankan perintah  “flutter pub upgrade” |
|  | Buat folder “repositories” pada folder “lib”, kemudian buat file “auth\_repo.dart”.  folder repositories ini akan kita gunakan untuk berkomunikasi dengan firebase/API dari back  end. |
|  | Kita akan membuat 2 buah Future<void> dalam class AuthRepo untuk login dan register pada file auth\_repo.dart. |
|  | Kemudian kita akan mengatur bloc untuk login. Buat folder dengan struktur seperti gambar di samping. |
|  | Tambahkan kode di samping pada file login\_state.dart |
|  | Tambahakan kode di samping pada file login\_cubit.dart |
|  | Jalankan program. inputan email dan password akan disimpan pada emailEdc dan passEdc kemudian  dengan menggunakan bloc inputan tersebut dikirim ke firebase melalui repositori yang sudah  kita buat, apabila berhasil maka akan masuk ke home screen, apabila gagal akan muncul  pesan gagal. |
|  | Mengatur untuk bagian register di firebase |
|  | Data yang baru ditambahkan, kita cek untuk mengisi pada bagian login. |
|  | Login sudah berhasil kita buat, maka selanjutnya kita akan membuat bagian register.  Karena kita sudah membuat Future<void> register. Selanjutnya kita atur untuk blocnya. Buat  folder “register” di dalam folder bloc. |
|  | Tambahkan kode di samping pada bagian file register\_state.dart |
|  | Tambahkan kode di samping pada bagian file register\_cubit.dart |
| *r* | Mencoba register |
|  | User yang baru didaftarkan berhasil ditambahkan |
|  | Mencoba login dengan akun yang baru saja di tambahkan |