

LAPORAN UJIAN TENGAH SEMESTER
PRATIUM PEMONGRAMAN SISTEM
BERGERAK

IdePlot – Firebase Api Login Google

202413430011



Oleh :

Rosul Iman (22343012)

Dosen Pengampu :

Geovanne Farell, S.P.d.,M.Pd.T.

S1 INFORMATIKA
DAPARTEMEN ELEKTRONIKA
UNIVERSITAS NEGERI PADANG

2024

A. KONSEP APLIKASI

IdePlot adalah aplikasi simple yang akan dikembangkan lebih lanjut, dalam laporan ini saya akan menjelaskan semua fungsi yang penting, dan memperlihatkan tampilan aplikasi ini.

IdePlot sendiri diambil dari dua kata yaitu “Ide” dan “Plot”, yang artinya aplikasi ini adalah aplikasi yang akan menampung Ide dan Plot dari sebuah cerita. Dalam aplikasi ini user atau pengguna akan melihat **tiga menu utama**, **Menu pertama** adalah menu untuk menambahkan plot dapat menambahkan plot, ada pilihan untuk plot, ada judul/nama plot, lalu deskripsi plot, kemudian ada poin-poin penting plot, pengguna dapat melihat plot tersebut dalam halaman berikutnya apabila menyimpannya, yang pada dasarnya tidak benar benar disimpan, hanya dikirim ke halaman viewPlot.

Selanjutnya pada **menu kedua** adalah menu untuk melihat plot, namun menu ini sebenarnya tidak akan berfungsi karena saya belum membuat system database di aplikasi ini.

Pada **menu ketiga** ada menu daftar, pengguna harus mengisi nama depan, nama belakang, email serta password, pada dasarnya informasi tidak disimpan ataupun diolah, hanya saja pada aplikasi ini nama depan akan diambil dan dikirimkan ke halaman utama lalu ditampilkan.

Untuk **fitur daftar dengan akun google**, pengguna akan memilih akun google yang telah login di hp pengguna, setelah pendaftaran berhasil, maka aplikasi akan mengambil nama depan, dan email kemudian menampilkan pesan berhasil dan kembali ke halaman utama, dan juga nama depan akan dipakai pada kata kata “Selamat datang, <nama pengguna>”

B. ALAT dan BAHAN

1. Laptop
2. Android Studio Sdk
3. Code Editor

C. TEORI SINGKAT

Firestore

Firestore adalah platform pengembangan aplikasi yang disediakan oleh Google untuk membantu developer dalam membuat aplikasi web dan mobile dengan cepat. Salah satu fitur utama Firestore adalah **Firestore Authentication**, yang menyediakan solusi untuk mengelola autentikasi pengguna dengan aman dan sederhana. Firestore Authentication ini mendukung berbagai metode login, seperti menggunakan email dan password, nomor telepon, atau login melalui platform pihak ketiga (OAuth) seperti Google, Facebook, Twitter, dan lain-lain.

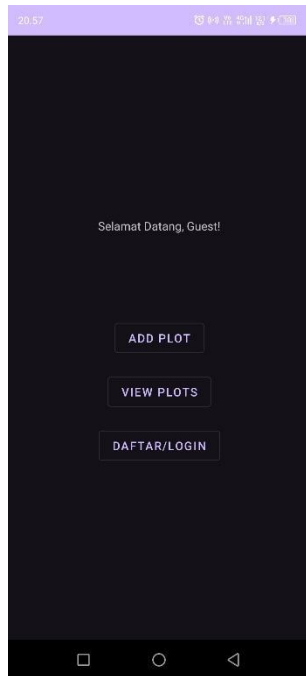
Komponen Firestore Authentication

1. **Login Email dan Password:** Firestore menyediakan fungsi autentikasi menggunakan email dan password secara langsung, di mana pengguna hanya perlu mendaftarkan email dan kata sandi mereka. Firestore mengelola penyimpanan data pengguna, sehingga developer tidak perlu membangun sistem autentikasi dari nol.
2. **OAuth untuk Platform Pihak Ketiga:** Firestore menyediakan API yang memungkinkan aplikasi terhubung dengan layanan login pihak ketiga, seperti Google Sign-In, Facebook Login, GitHub, dan lain-lain. Hal ini memudahkan pengguna untuk login menggunakan akun yang sudah ada di platform lain, tanpa perlu membuat akun baru.
3. **Custom Token:** Jika Anda memiliki sistem autentikasi yang kompleks atau pengguna di berbagai platform dengan autentikasi unik, Firestore memungkinkan Anda untuk membuat token khusus (custom token) yang bisa digunakan dalam Firestore Authentication. Custom token memungkinkan developer untuk membuat aturan autentikasi yang disesuaikan dengan kebutuhan aplikasi.
4. **Integrasi API di Berbagai Platform:** Firestore menyediakan SDK dan API yang bisa digunakan di berbagai platform, seperti Android, iOS, dan aplikasi web. Dengan ini, developer bisa mengintegrasikan Firestore Authentication dengan mudah ke dalam aplikasi mereka, memungkinkan pengguna login di berbagai perangkat dan platform dengan akun yang sama.

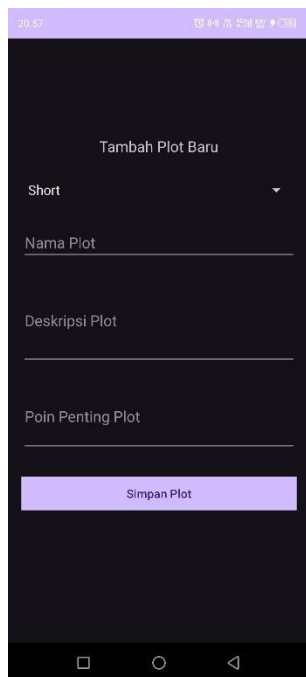
D. LANGKAH KERJA

a) Menyiapkan Tampilan Aplikasi

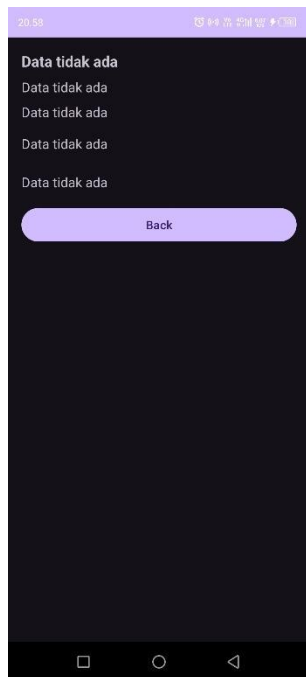
1. Mendesain tampilan **Home**



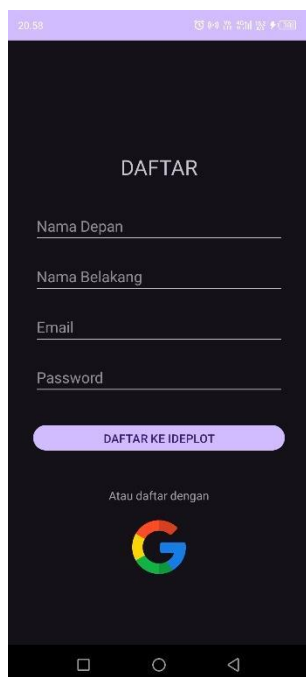
2. Mendesain tampilan **Halaman AddPlot**



3. Mendesain tampilan Halaman ViewPlot

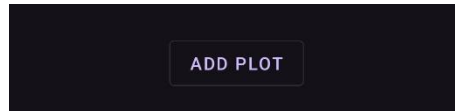


4. Mendesain tampilan Halaman Daftar/Login



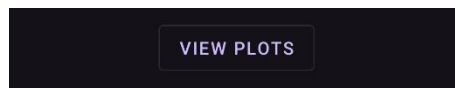
b) Menyiapkan Code Tombol Beralih Halaman

1. Halaman Home ke Halaman AddPlot



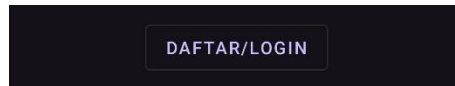
```
// Handle "Add Plot" button click (jika diperlukan)
val btnAddPlot: MaterialButton = findViewById(R.id.btnAddPlot)
btnAddPlot.setOnClickListener {
    val intent = Intent(this, AddPlotActivity::class.java)
    startActivity(intent)
}
```

2. Halaman Home ke Halaman ViewPlot



```
// Handle "View Plot" button click
val btnViewPlot: MaterialButton = findViewById(R.id.btnViewPlot)
btnViewPlot.setOnClickListener {
    val intent = Intent(this, ViewPlotActivity::class.java)
    startActivity(intent)
}
```

3. Halaman Home ke Halaman Daftar/Login



```
// Handle "Btn Login" button click
val btnLogin: MaterialButton = findViewById(R.id.btn4)
btnLogin.setOnClickListener {
    val intent = Intent(this, HalamanDaftar::class.java)
    startActivity(intent)
}
```

4. Halaman AddPlot ke Halaman ViewPlot

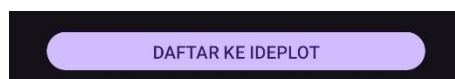


Fungsi savePlot akan mengarahkan ke halaman viewPlot dengan mengirimkan data input dari halaman AddPlot, fungsi ini akan lebih dijelaskan lebih lanjut dipoin selanjutnya.

```
// Set click listener for the save button
btnSavePlot.setOnClickListener {
    savePlot()
}
```

5. Halaman Daftar/Login ke Halaman Home

Fungsi Tombol Daftar Biasa (tanpa Google)



Fungsi ini juga menangani hasil input dari halaman Daftar/Login, hasil input ini akan diambil nama depan, lalu di kirim ke halaman Home, dan juga untuk menampilkan pesan bahwa pendaftaran berhasil.

```
findViewById<Button>(R.id.BtnDaftar).setOnClickListener {
    val namaDepan = editTextNamaDepan.text.toString().trim()
    val namaBelakang = editTextNamaBelakang.text.toString().trim()
    val email = editTextEmail.text.toString().trim()
    val password = editTextPassword.text.toString().trim()

    // Validasi input
    if (namaDepan.isEmpty() || namaBelakang.isEmpty() ||
    email.isEmpty() || password.isEmpty()) {
        Toast.makeText(this, "Harap isi semua kolom",
        Toast.LENGTH_SHORT).show()
    } else {
        // Proses pendaftaran
        val message = "Pendaftaran Berhasil!\nNama: $namaDepan
        $namaBelakang\nEmail: $email"
        Toast.makeText(this, message, Toast.LENGTH_LONG).show()

        // Pindah ke MainActivity dengan membawa isi dari variabel
        nama depan
        val intent = Intent(this, MainActivity::class.java)
        intent.putExtra("NAMA_DEPAN", namaDepan) // Mengirim nama
        depan
        startActivity(intent)

        // Menutup activity ini agar tidak kembali ke sini setelah
        menekan back
        finish()
    }
}
```

Fungsi Tombol Daftar dengan Google



Tombol pendaftaran google hanya memanggil fungsi **signInWithGoogle()** setiap kali ditekan, setiap kali tombol ini ditekan, akan menghapus akses akun dari aplikasi, kemudian logout dan memulai proses login lagi. Fungsi **signInWithGoogle** akan dijelaskan lebih lanjut nanti.

```
// Panggil fungsi login saat tombol ditekan
findViewById<AppCompatActivity>(R.id.btnLoginGoogle).setOnClickListener {
    googleSignInClient.revokeAccess() // Hapus akses akun
    .addOnCompleteListener {
        googleSignInClient.signOut() // Logout dari akun Google
        .addOnCompleteListener {
            signInWithGoogle() // Mulai ulang proses login Google
        }
    }
}
```

c) Menyiapkan Code Fitur Pada Menu AddPlot (Menu 1)

Pada halaman atau menu AddPlot terdapat satu fungsi yang dipanggil pada tombol AddPlot ke ViewPlot yaitu **savePlot()**, Berikut penjelasannya.

Fungsi savePlot()

Fungsi ini akan bertanggung jawab atas hasil view pada halaman viewPlot, setelah mengambil input dari pengguna, fungsi ini akan mengirimkan data tersebut ke halaman viewPlot

```
// Fungsi savePlot
private fun savePlot() {
    val plotLength = spinnerPlotLength.selectedItem.toString()
    val plotName = editTextPlotName.text.toString().trim()
    val plotDescription = editTextPlotDescription.text.toString().trim()
    val importantPoints = editTextImportantPoints.text.toString().trim()

    if (plotName.isEmpty() || plotDescription.isEmpty() ||
        importantPoints.isEmpty()) {
        Toast.makeText(this, "Harap isi semua kolom",
            Toast.LENGTH_SHORT).show()
        return
    }

    // Mendapatkan tanggal saat ini secara dinamis
    val dateFormat = SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())
    val plotDate = dateFormat.format(java.util.Date())

    // Membuat Intent untuk memulai ViewPlotActivity dan mengirim data
    plot
    val intent = Intent(this, ViewPlotActivity::class.java).apply {
        putExtra("PLOT_NAME", plotName)
        putExtra("PLOT_LENGTH", plotLength)
        putExtra("PLOT_DATE", plotDate) // Mengirim tanggal sebagai
    string
        putExtra("PLOT_DESCRIPTION", plotDescription) // Mengirim
    deskripsi plot
        putExtra("PLOT_POINTS", importantPoints) // Mengirim poin penting
    plot
    }
    startActivity(intent)

    // Menghapus isi kolom setelah menyimpan
    clearInputFields()
}
```

Fungsi clearInputFields()

Selanjutnya ada fungsi lain yang dipanggil dari fungsi savePlot() yaitu fungsi **clearInputFields()** yang berfungsi untuk mengclear input pengguna.

```
// Function to clear input fields after saving
private fun clearInputFields() {
    editTextPlotName.text.clear()
    editTextPlotDescription.text.clear()
    editTextImportantPoints.text.clear()
    spinnerPlotLength.setSelection(0)
}
```

d) Menyiapkan Code Fitur Pada Menu ViewPlot (Menu 2)

Kode untuk mengambil data yang dikirim dari halaman addPlot.

Kode dibawah ini menginisialisasi tampilan elemen-elemen `TextView` dan tombol `btnBack`, lalu mengambil data dari `Intent` yang dikirim dari `AddPlotActivity`. Data yang diambil meliputi nama plot, panjang, tanggal, deskripsi, dan poin penting plot. Jika salah satu data kosong, teks "Data tidak ada" akan ditampilkan pada setiap `TextView`. Jika semua data tersedia, data tersebut akan diatur pada `TextView` sesuai dengan nilai yang diterima.

```
// Inisialisasi tampilan
textViewPlotName = findViewById(R.id.textViewPlotName)
textViewPlotLength = findViewById(R.id.textViewPlotLength)
textViewPlotDate = findViewById(R.id.textViewPlotDate)
textViewPlotDescription = findViewById(R.id.textViewPlotDescription)
textViewPlotPoints = findViewById(R.id.textViewPlotPoints)
btnBack = findViewById(R.id.btnBack)

// Mendapatkan data plot dari intent (dikirim dari AddPlotActivity)
val plotName = intent.getStringExtra("PLOT_NAME")
val plotLength = intent.getStringExtra("PLOT_LENGTH")
val plotDate = intent.getStringExtra("PLOT_DATE") // Mengasumsikan tanggal
dikirim sebagai string
val plotDescription = intent.getStringExtra("PLOT_DESCRIPTION") // Deskripsi plot
val plotPoints = intent.getStringExtra("PLOT_POINTS") // Poin penting dari plot

// Validasi dan menampilkan data yang diterima ke tampilan text view
masing-masing
if (plotName.isNullOrEmpty() || plotLength.isNullOrEmpty() ||
    plotDate.isNullOrEmpty() || plotDescription.isNullOrEmpty() || plotPoints.isNullOrEmpty()) {
    // Menampilkan pesan "Data tidak ada" jika salah satu data tidak tersedia
    textViewPlotName.text = "Data tidak ada"
    textViewPlotLength.text = "Data tidak ada"
    textViewPlotDate.text = "Data tidak ada"
    textViewPlotDescription.text = "Data tidak ada"
    textViewPlotPoints.text = "Data tidak ada"
} else {
    // Mengatur data yang diterima ke tampilan text view masing-masing
    textViewPlotName.text = plotName
    textViewPlotLength.text = plotLength
    textViewPlotDate.text = plotDate
    textViewPlotDescription.text = plotDescription
    textViewPlotPoints.text = plotPoints
}
```

e) Menyiapkan Code Fitur Pada Menu Daftar/Login (Menu 3)

Kode untuk mendaftar, pada dasarnya fungsi pendaftaran ini saya satukan dengan tombol daftar ketika ditekan. Berikut adalah fungsi untuk menangani pendaftaran pengguna jika tidak menggunakan akun google.

Kode untuk menangani daftar tanpa akun google

```
// Jika USER DAFTAR dengan NAMA DAN EMAIL BIASA tanpa akun
// Menangani klik tombol "Daftar"

editTextNamaDepan = findViewById(R.id.editTextNamaDepan)
editTextNamaBelakang = findViewById(R.id.editTextNamaBelakang)
editTextEmail = findViewById(R.id.editTextTextEmailAddress)
editTextPassword = findViewById(R.id.editTextTextPassword)

findViewById<Button>(R.id.BtnDaftar).setOnClickListener {
    val namaDepan = editTextNamaDepan.text.toString().trim()
    val namaBelakang = editTextNamaBelakang.text.toString().trim()
    val email = editTextEmail.text.toString().trim()
    val password = editTextPassword.text.toString().trim()

    // Validasi input
    if (namaDepan.isEmpty() || namaBelakang.isEmpty() || email.isEmpty() ||
password.isEmpty()) {
        Toast.makeText(this, "Harap isi semua kolom", Toast.LENGTH_SHORT).show()
    } else {
        // Proses pendaftaran
        val message = "Pendaftaran Berhasil!\nNama: $namaDepan
$namaBelakang\nEmail: $email"
        Toast.makeText(this, message, Toast.LENGTH_LONG).show()

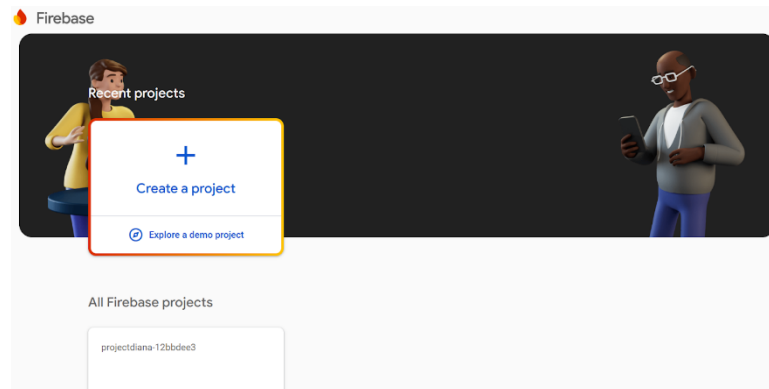
        // Pindah ke MainActivity dengan membawa isi dari variabel nama depan
        val intent = Intent(this, MainActivity::class.java)
        intent.putExtra("NAMA_DEPAN", namaDepan) // Mengirim nama depan
        startActivity(intent)

        // Menutup activity ini agar tidak kembali ke sini setelah menekan back
        finish()
    }
}
```

Kode diatas menangani pendaftaran pengguna dengan nama dan email tanpa akun eksternal. Tampilan input untuk nama depan, nama belakang, email, dan password diinisialisasi, lalu tombol daftar disiapkan untuk memvalidasi input ketika ditekan. Jika ada kolom kosong, pesan peringatan akan ditampilkan. Jika semua data terisi, sistem menampilkan pesan keberhasilan, mengirim data namaDepan ke MainActivity melalui Intent, dan menutup halaman saat ini agar tidak kembali setelah menekan tombol "back."

f) Membuat API untuk Pendaftaran Google dengan Firebase

1. Login Firebase dan Buat Project



Buatlah project > lalu pada menu untuk memasukkan kode SHA1 boleh diskrip ataupun dimasukkan, berikut cara untuk mendapatkan kode SHA1 pada project.

Buat kode SHA1 di terminal android studio

code : **gradlew signingReport**

dapat dijalankan di terminal di android Studio jika JAVA_HOME nya sudah di set terlebih dahulu.

Atau jalankan di direktori openjdk default dari android studio

Code : `keytool -list -v -keystore <user-home>\.android\debug.keystore`

Password : android

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Iman874>cd C:\My Program Files\Android_Studio\jbr\bin

C:\My Program Files\Android_Studio\jbr\bin>keytool -list -v -keystore C:\\Users\\Iman874\\.android\\debug.keystore
Enter keystore password:
Keystore type: PKCS12
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: androiddebugkey
Creation date: 27 Okt 2024
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, CN=Android Debug
Issuer: C=US, O=Android, CN=Android Debug
Serial number: 1
Valid from: Sun Oct 27 15:30:07 WIB 2024 until: Tue Oct 20 15:30:07 WIB 2054
Certificate fingerprints:
  SHA1: FD:30:C3:2E:3E:B1:3F:F3:84:4E:52:D6:4C:7E:70:28:84:5A:31:21
  SHA256: 6B:EF:4A:10:95:B6:7C:BB:08:64:F9:3F:E7:D4:31:0E:54:C4:12:94:CD:66:19:B1:DC:44:67:76:2B:21:FA:10
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 1

*****
```

2. Masukkan kode SHA1 pada project firebase

 **IdePlot**
com.example.ideplot

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

[See SDK instructions](#) [google-services.json](#)

App ID ⓘ
1:922977622414:android:be1c120f7b13a8026decdf

App nickname
IdePlot ✎

Package name
com.example.ideplot

SHA certificate fingerprints ⓘ
fd:30:c3:2e:3e:b1:3f:f3:84:4e:52:d6:4c:7e:70:28:84:5a:31:21

Type ⓘ
SHA-1

[Add fingerprint](#)

3. Konfigurasi Plugin, dan Dependensi di Android Studio

Tambahkan Plugin di Build Gradle (Project)

```
id("com.google.gms.google-services") version "4.4.2" apply false
```

Tambahkan Plugin di Build Gradle (APP)

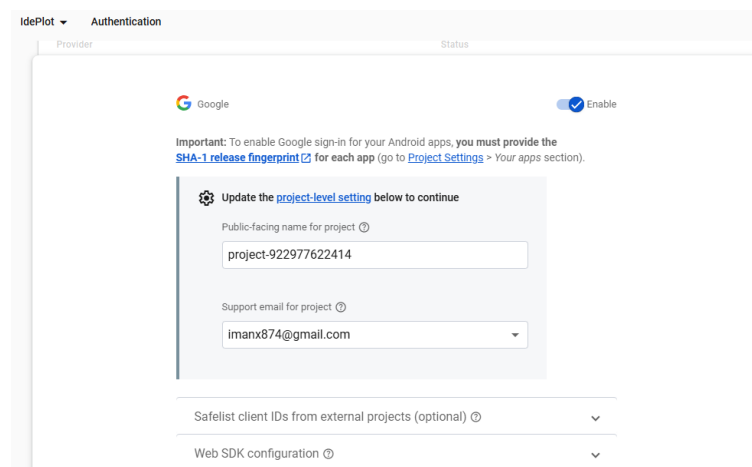
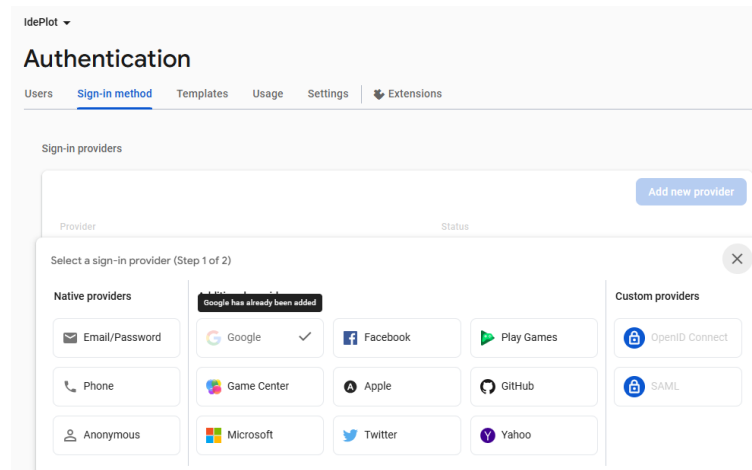
```
id("com.google.gms.google-services")
```

Tambahkan Dependensi yang diperlukan di Build Gradle (APP)

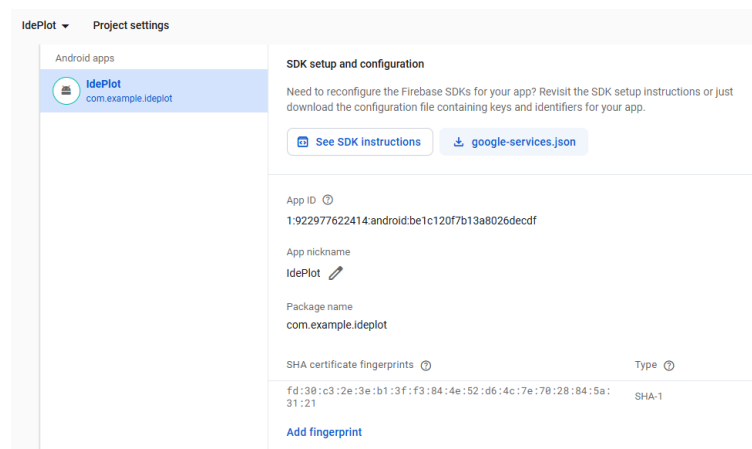
```
// Import the Firebase dependensi
implementation(platform("com.google.firebase:firebase-bom:33.5.1"))
implementation("com.google.firebase:firebase-analytics")
implementation("com.google.firebase:firebase-firestore")

// Import google libbary autentifikasi
implementation("com.google.android.gms:play-services-auth:21.2.0")
```

4. Buat Metode Login dan pasang file konfigurasi



Pasang file konfigurasi **google-services.json** pada direktori project.



Sync project dan build project

g) Menyiapkan Code Untuk Fitur Pendaftaran Dengan Google

Deklarasi Variabel yang diperlukan

```
// Tambahkan variabel untuk Google Sign-In
private lateinit var googleSignInClient: GoogleSignInClient
private lateinit var auth: FirebaseAuth
```

Deklarasi Code di Oncreate untuk Login

```
// Konfigurasi opsi login Google
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestEmail()
    .build()

googleSignInClient = GoogleSignIn.getClient(this, gso)
auth = FirebaseAuth.getInstance()
```

Kode diatas mengonfigurasi opsi untuk login dengan Google menggunakan GoogleSignInOptions. Opsi login diatur dengan DEFAULT_SIGN_IN untuk mendapatkan ID token dan alamat email pengguna, menggunakan requestIdToken dengan ID klien web yang diambil dari sumber daya string aplikasi. Kemudian, objek googleSignInClient diinisialisasi untuk mengelola proses login Google, dan auth diatur menggunakan FirebaseAuth.getInstance() untuk menghubungkan autentikasi Google dengan Firebase, memungkinkan login dan akses aman ke fitur

Fungsi untuk Login signInWithGoogle()

```
// Fungsi untuk memulai Google Sign-In
private fun signInWithGoogle() {
    val signInIntent = googleSignInClient.signInIntent
    googleSignInLauncher.launch(signInIntent)
}
```

Kode diatas mendefinisikan fungsi signInWithGoogle untuk memulai proses login dengan Google. Fungsi ini membuat signInIntent menggunakan googleSignInClient, yang mengarahkan pengguna ke antarmuka login Google. Setelah itu, googleSignInLauncher.launch(signInIntent) memulai aktivitas login, memungkinkan pengguna untuk memilih akun Google mereka dan memberikan autentikasi yang diperlukan bagi aplikasi.

Instance variabel googleSignInLauncher

Instance ini berfungsi untuk dipanggil di fungsi signInWithGoogle()

```
private val googleSignInLauncher = registerForActivityResult(
    ActivityResultContracts.StartActivityForResult()
) { result ->
    val task = GoogleSignIn.getSignedInAccountFromIntent(result.data)
```

```

        if (task.isSuccessful) {
            val account = task.result
            if (account != null) {
                firebaseAuthWithGoogle(account)
            }
        } else {

            // Mengambil informasi error secara detail
            val errorCause = task.exception?.cause
            val errorClass = task.exception?.javaClass?.simpleName
            val errorStackTrace = task.exception?.stackTrace?.joinToString("\n")

            // Menampilkan pesan error dari Google Sign-In
            val errorMessage = task.exception?.message

            Log.e("GoogleSignInError", "Google Sign-In gagal dengan error:")
            Log.e("GoogleSignInError", "Error Message: $errorMessage")
            Log.e("GoogleSignInError", "Error Cause: $errorCause")
            Log.e("GoogleSignInError", "Error Class: $errorClass")
            Log.e("GoogleSignInError", "Stack Trace: $errorStackTrace")
            // Menampilkan pesan Toast dengan pesan error jika ada, atau pesan
            default jika errorMessage null
            Toast.makeText(this, errorMessage ?: "Terjadi kesalahan pada Google
            Sign-In", Toast.LENGTH_LONG).show()
        }
    }
}

```

Kode diatas mendeklarasikan `googleSignInLauncher` untuk menangani hasil dari proses Google Sign-In dengan `registerForActivityResult`. Setelah pengguna menyelesaikan proses login, hasilnya diterima dalam `result`, dan `task` memeriksa apakah login berhasil. Jika sukses, `firebaseAuthWithGoogle` dipanggil untuk mengautentikasi akun dengan Firebase. Jika login gagal, berbagai informasi error—termasuk pesan error, penyebab, kelas error, dan stack trace—dicatat di log untuk membantu debugging. Selain itu, pesan `Toast` ditampilkan kepada pengguna, menampilkan pesan error atau pesan default jika `errorMessage` null.

Fungsi `firebaseAuthWithGoogle(account: GoogleSignInAccount)`

yang berguna untuk autentikasi pengguna ke Firebase menggunakan kredensial yang diperoleh dari akun Google, fungsi ini dipanggil di instance variabel **`googleSignInLauncher`**

```

private fun firebaseAuthWithGoogle(account: GoogleSignInAccount) {
    val credential = GoogleAuthProvider.getCredential(account.idToken, null)
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                try {
                    val namaDepan = account.givenName ?: "Guest"
                    val namaBelakang = account.familyName ?: ""
                    val email = account.email ?: ""

                    val message = "Pendaftaran Berhasil!\nNama: $namaDepan
                    $namaBelakang\nEmail: $email"
                    Toast.makeText(this, message, Toast.LENGTH_LONG).show()
                } catch (e: Exception) {
                    Log.e("FirebaseAuth", "Error during authentication: $e")
                }
            } else {
                Log.e("FirebaseAuth", "Authentication failed: ${task.exception?.message}")
            }
        }
}

```

```

        val intent = Intent(this, MainActivity::class.java)
        intent.putExtra("NAMA_DEPAN", namaDepan)
        startActivity(intent)
        finish()
    } catch (e: ApiException) {
        Log.w("HalamanDaftar", "signInResult:failed code=" +
e.statusCode)
        Toast.makeText(this, "Login Google gagal",
Toast.LENGTH_SHORT).show()
    }
    } else {
        val errorCode = task.exception?.message
        Log.e("FirebaseAuthError", "Firebase Auth gagal: $errorCode",
task.exception)
    }
}
}

```

Kode diatas mendefinisikan fungsi firebaseAuthWithGoogle untuk mengautentikasi akun Google menggunakan Firebase. Fungsi ini menerima account yang diperoleh dari Google Sign-In dan membuat credential menggunakan token ID akun. Jika autentikasi berhasil, kode mengekstrak informasi pengguna seperti nama depan, nama belakang, dan email. Pesan sukses ditampilkan melalui Toast, dan pengguna diarahkan ke MainActivity dengan data namaDepan yang diteruskan. Jika ada kegagalan dalam proses, informasi error dicatat di log, dan pesan error ditampilkan kepada pengguna.

E. TES APLIKASI

Video testing : <https://youtu.be/krbCXqPlja8?si=C3vMxuuVIEzYb8w4>

Source code project : [GitHub - Iman874/IdePlot-Firebase-API-Login-Google](#)