



Institut Teknologi Telkom Purwokerto 2020/2021

PEMROGRAMAN PERANGKAT BERGERAK

Modul Praktikum Topik 11

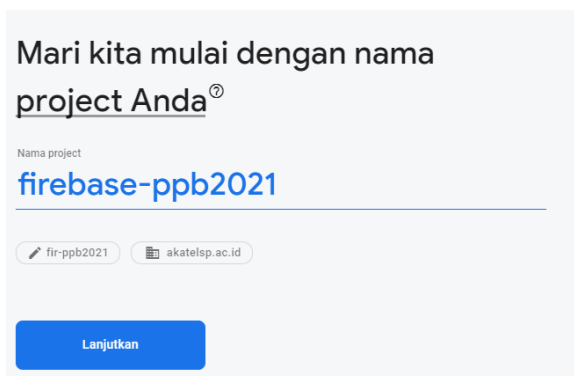
- Konfigurasi Firebase
- Auth Email/Password
- Auth Google

11.1 Persiapan

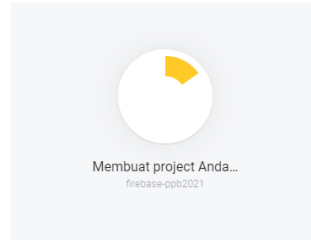
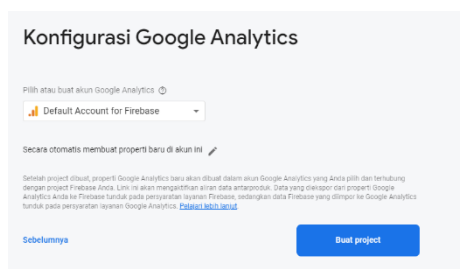
1. Android Studio
2. Device Android/Emulator Android
3. Kabel Data
4. Buat module praktikum 11

11.2 Konfigurasi Firebase

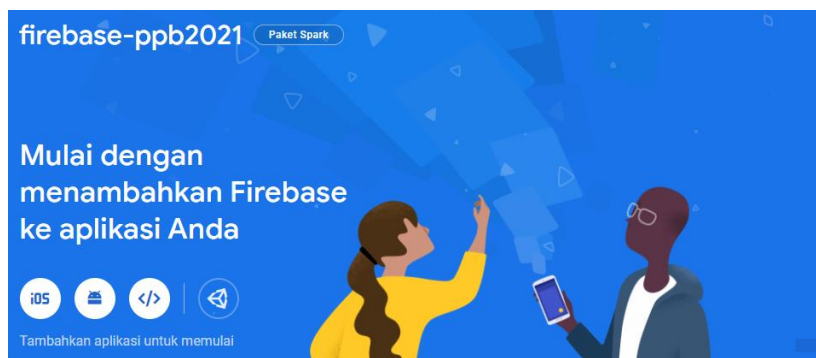
1. Pada modul praktikum ini akan menggunakan akun goggle milik institusi yaitu @ittelkom-pwt.ac.id. Jika ada yang mengalami kendala pada akun tersebut dapat menggunakan akun @google.com.
2. Buka <https://console.firebase.google.com/>. Pada modul praktikum ini bahasa yang digunakan pada <https://console.firebase.google.com/> adalah bahasa Indonesia.
3. Pilih **tambahkan project**, lalu masukan nama project yang akan kita buat. Kita samakan dengan nama **firebase-ppb2021**.



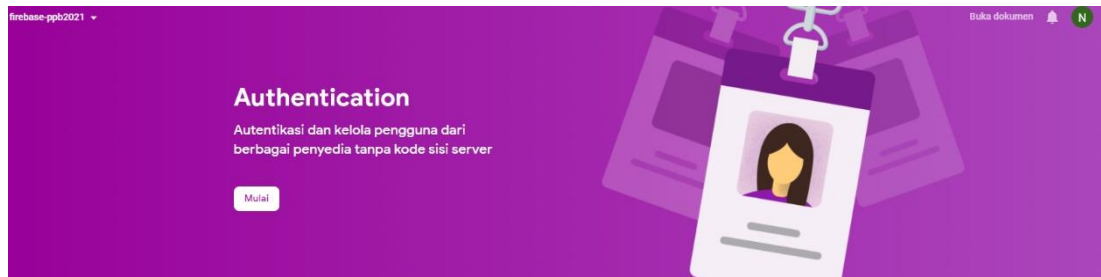
4. Tekan tombol lanjutkan sampai menemukan tombol **Buat project**.



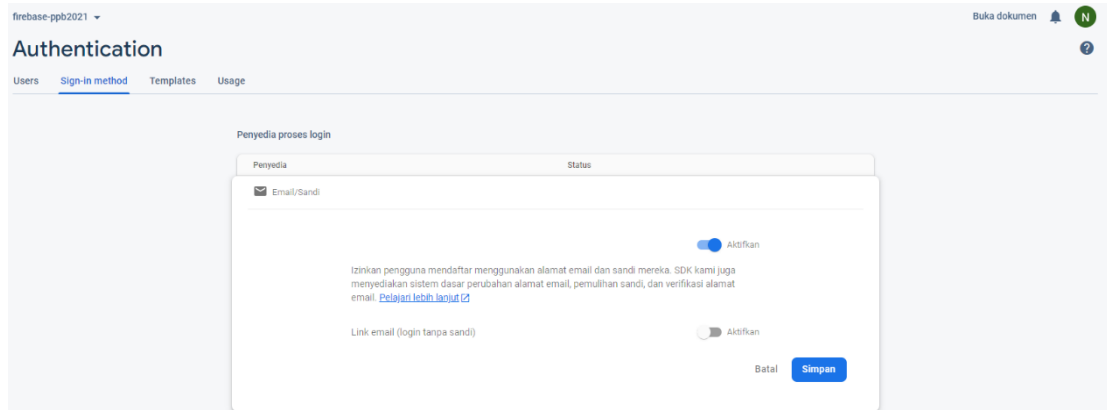
5. Sampai tahap ini project firebase kita telah berhasil dibuat.



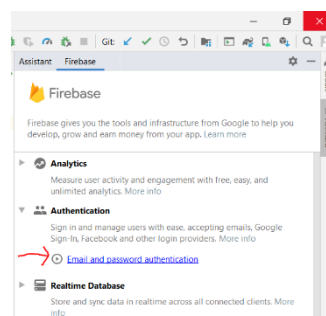
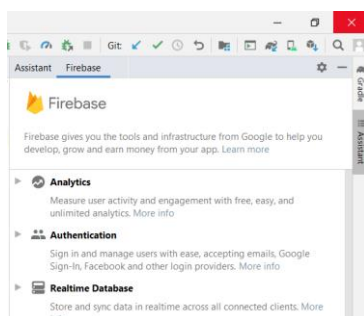
- Pilih menu **Authentication**, lalu pilih tombol mulai.



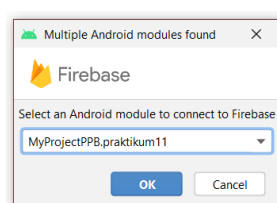
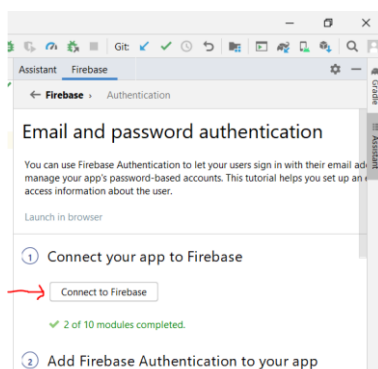
- Aktifkan Email/Sandi.



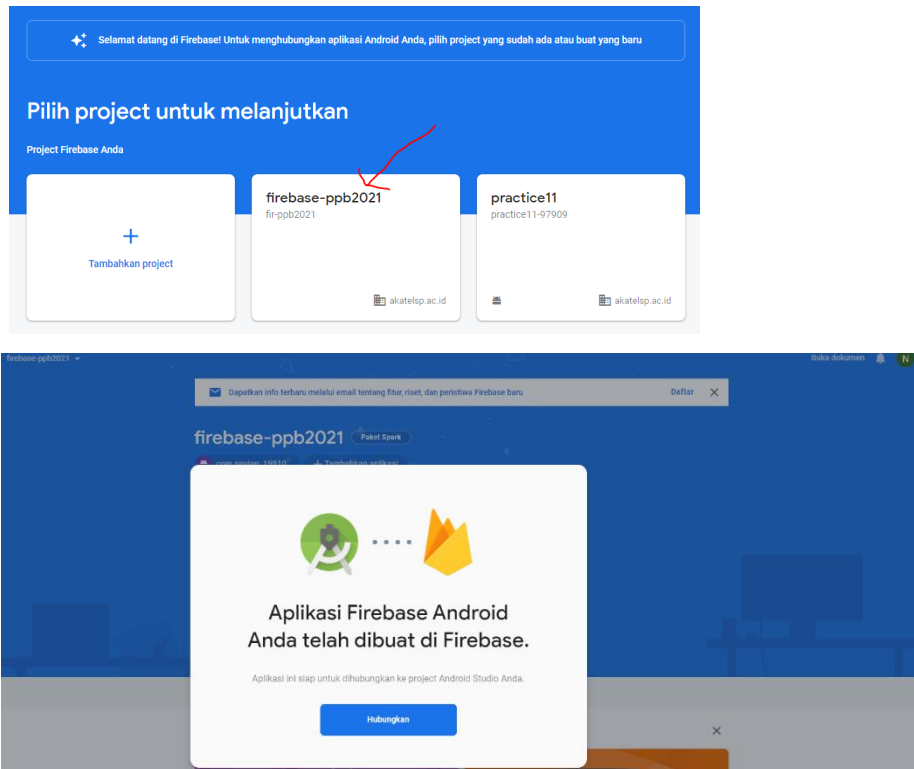
- Sampai tahap ini persiapan Authentication menggunakan Email/Sandi sudah siap.
- Buka project android studio, lalu buka pada bagian module praktikum11.
- Buka tab firebase dengan memilih menu tools/firebase. Maka akan tampil tab di sebelah kanan seperti berikut ini. Buka **Authentication** lalu klik **Email and password authentication**.



- Klik Connect to Firebase.



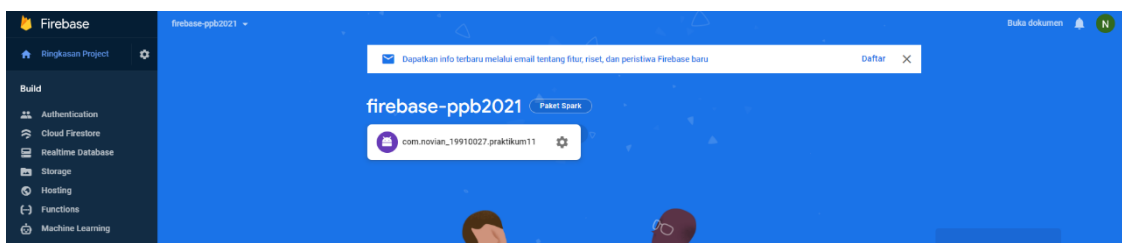
12. Seketika browser akan terbuka dan kita diharuskan memilih project firebase yang akan ditautkan dengan project android kita.



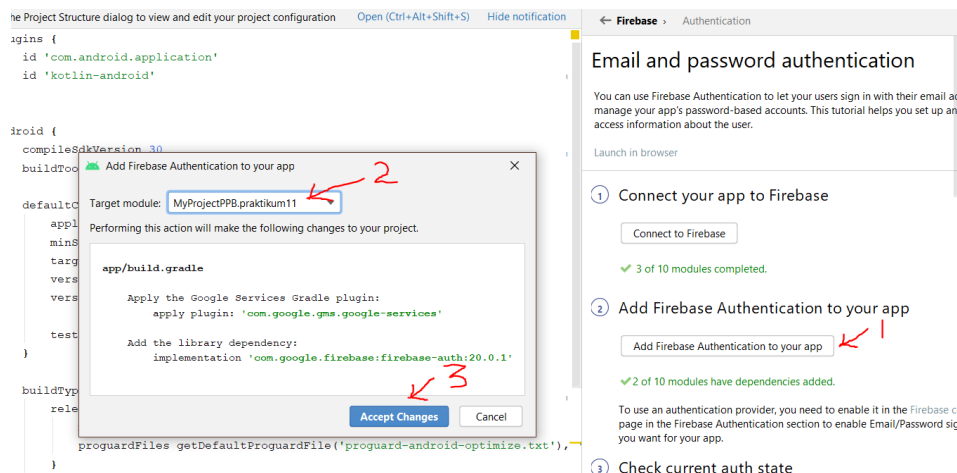
Your Android Studio project is connected to your Firebase Android app

You can now use Firebase in your project! Go back to Android Studio to start using one of the Firebase SDKs.

13. Sampai tahap ini firebase telah tertaut dengan project android kita, hal ini bisa dilihat pada laman project firebase terdapat nama package aplikasi milik kita.



14. Tahap akhir pada bagian ini adalah selanjutnya kita kembali ke android studio lalu lanjutkan langkah kedua pada tab firebase.

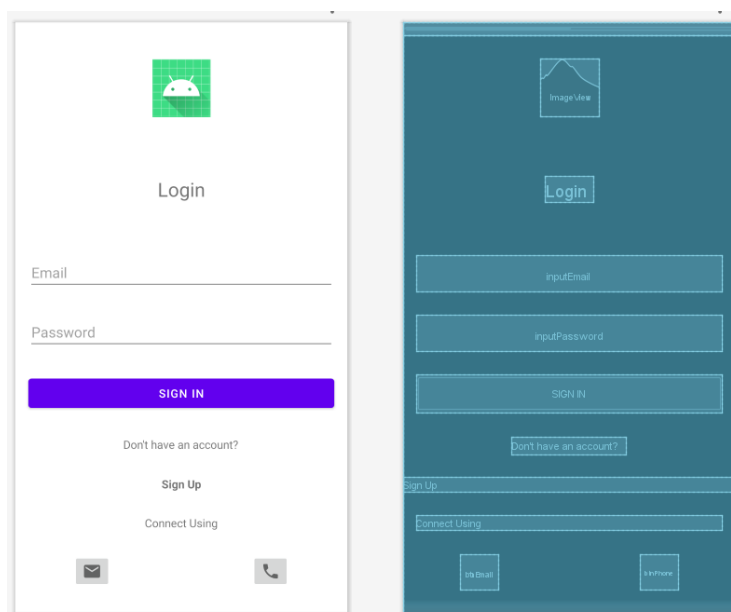


11.3 Auth Email/Sandi

1. Pada gradle tambahkan dependensi berikut ini.

```
implementation platform('com.google.firebase:firebase-bom:26.2.0')
implementation 'com.google.firebase:firebase-auth-ktx'
```

2. Buat activity baru dengan nama **SignInActivity** dan **SignUpActivity**.
3. Buka **activity_sign_in.xml** tambahkan kode berikut ini ([link](#)), kita tidak akan mengulas isi dari **activity_sign_in.xml** karena desain layout sudah kita pelajari cukup banyak selama praktikum sebelumnya. Inti dari **activity_sign_in.xml** ini adalah kita akan menggunakan **btnSign** untuk melakukan proses login menggunakan email/sandi, untuk membuka aktivitas sign up, **btnEmail** untuk login menggunakan google, **btnPhone** untuk login menggunakan nomor handphone, **inputEmail** untuk mengisi email yang akan digunakan untuk login, **inputPassword** untuk mengisi sandi yang akan digunakan untuk login.



4. Buka **SignInActivity.kt**, ubah dan tambahkan kode seperti dibawah ini.

```
class SignInActivity : AppCompatActivity(), View.OnClickListener {
    private lateinit var auth: FirebaseAuth
    private lateinit var binding: ActivitySignInBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivitySignInBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btnSign.setOnClickListener(this)
        binding.tvSignUp.setOnClickListener(this)
        binding.btnEmail.setOnClickListener(this)
        binding.btnPhone.setOnClickListener(this)

        auth = FirebaseAuth

    }

    override fun onClick(v: View) {
        TODO("Not yet implemented")
    }
}
```

Pada tahap ini kita telah menerapkan binding beserta onclick pada masing-masing komponen yang ada di **activity_sign_in.xml**. Kita juga telah melakukan deklarasi untuk FirebaseAuth.

5. Pada fungsi onClick kita tambahkan kode berikut ini.

```
override fun onClick(v: View) {
    when (v.id) {
        R.id.btnSign -> {
            signIn(binding.inputEmail.text.toString(),
                binding.inputPassword.text.toString())
        }
        R.id.tvSignUp -> {
            val intent = Intent(this@SignInActivity, SignUpActivity::class.java)
            startActivity(intent)
        }
    }
}
```

6. Pada kode nomor 5 tombol **btnSign** diperintahkan untuk mengeksekusi fungsi signIn() namun fungsi tersebut masih belum dibuat, tambahkan fungsi **signIn()** seperti dibawah ini.

```
private fun signIn(email: String, password: String) {
    if (!validateForm()) {
        return
    }
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                finish()
                val intent = Intent(this@SignInActivity,
                    MainActivity::class.java)
                startActivity(intent)
            } else {
                Toast.makeText(baseContext, "Authentication failed.",
                    Toast.LENGTH_SHORT).show()
            }
        }
}
```

7. Buka **activity_main.xml** tambahkan kode berikut ini ([link](#)), kita tidak akan mengulas isi dari **activity_main.xml** karena desain layout sudah kita pelajari cukup banyak selama praktikum sebelumnya. Inti dari **activity_main.xml** ini adalah terdapat **btnEmailVerify** untuk mengirim email berupa verifikasi, **btnSignOut** untuk melakukan proses sign out, **tvName** untuk menampilkan nama user, **tvUserId** untuk menampilkan id user.
8. Buka **MainActivity.kt** lalu tambahkan kode berikut ini.

```
class MainActivity : AppCompatActivity() {
    private lateinit var auth: FirebaseAuth
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        auth = Firebase.auth

        val currentUser = auth.currentUser
        if (currentUser == null) {
            val intent = Intent(this@MainActivity, SignInActivity::class.java)
            startActivity(intent)
            finish()
        }
    }
    public override fun onStart() {
        super.onStart()
        val currentUser = auth.currentUser
        if (currentUser == null) {
            val intent = Intent(this@MainActivity, SignInActivity::class.java)
            startActivity(intent)
            finish()
        }
    }
}
```

Kode diatas digunakan untuk memvalidasi ketika user belum login maka **MainActivity** tidak dapat dibuka.

9. Masih pada **MainActivity.kt**, kita tambahkan onclick pada tombol **btnEmailVerify** dan **btnSignOut**.

```
startActivity(intent)
finish()
}
binding.btnSignOut.setOnClickListener(this)
binding.btnEmailVerify.setOnClickListener(this)
}
```

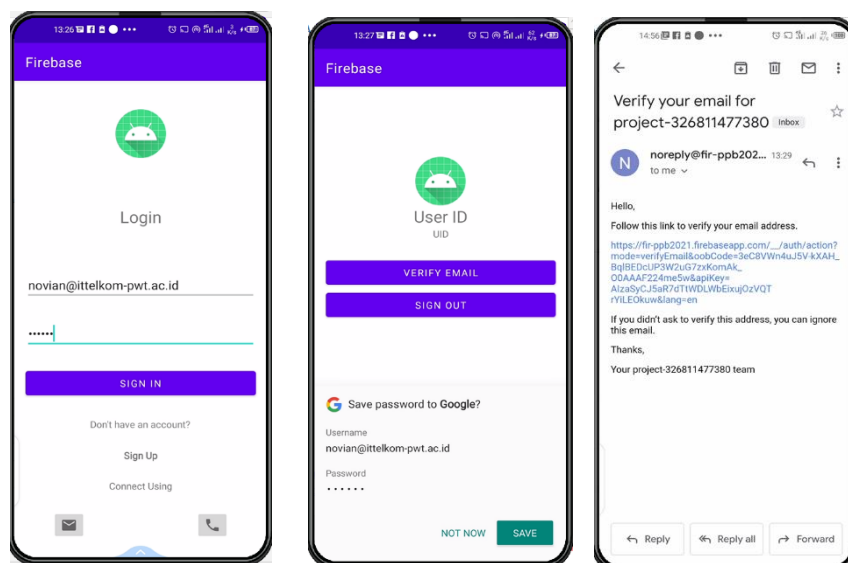
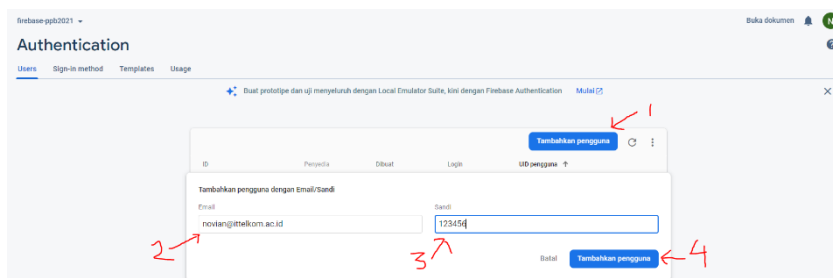
```
override fun onClick(v: View) {
    when (v.id) {
        R.id.btnSignOut -> {
            signOut()
        }
        R.id.btnEmailVerify -> {
            sendEmailVerification()
        }
    }
}
```

10. Onclick sudah berhasil di tambahkan, selanjutnya kita akan menambahkan masing-masing fungsi pada tombol tersebut.

```
private fun sendEmailVerification() {
    binding.btnEmailVerify.isEnabled = false
    val user = auth.currentUser!!
    user.sendEmailVerification()
        .addOnCompleteListener(this) { task ->
            binding.btnEmailVerify.isEnabled = true
            if (task.isSuccessful) {
                Toast.makeText(baseContext,
                    "Verification email sent to ${user.email} ",
                    Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(baseContext,
                    "Failed to send verification email.",
                    Toast.LENGTH_SHORT).show()
            }
        }
}

private fun signOut() {
    auth.signOut()
    val currentUser = auth.currentUser
    if (currentUser == null) {
        val intent = Intent(this@MainActivity, SignInActivity::class.java)
        startActivity(intent)
        finish()
    }
}
```

11. Pada tahap ini proses login, logout dan email verifikasi sudah dapat dilakukan. Sebelum uji coba kita perlu membuat secara manual melalui firebase karena fitur create akun masih belum kita buat.

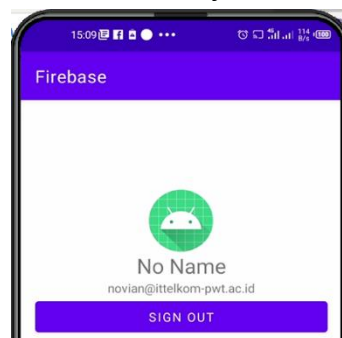


12. Pada tahap di atas kita lihat pengujian telah berhasil dilakukan, namun kita lihat nama akun masih belum tampil. Buka **MainActivity.kt** tambahkan kode berikut ini.

```
private fun updateUI(currentUser: FirebaseUser) {
    currentUser?.let {
        val name = currentUser.displayName
        val phoneNumber = currentUser.phoneNumber
        val email = currentUser.email
        val photoUrl = currentUser.photoUrl
        val emailVerified = currentUser.isEmailVerified
        val uid = currentUser.uid
        binding.tvName.text = name
        if(TextUtils.isEmpty(name)){
            binding.tvName.text = "No Name"
        }
        binding.tvUserId.text = email
        for (profile in it.providerData) {
            val providerId = profile.providerId
            if(providerId=="password" && emailVerified==true){
                binding.btnEmailVerify.isVisible = false
            }
            if(providerId=="phone"){
                binding.tvName.text = phoneNumber
                binding.tvUserId.text = providerId
            }
        }
    }
}

public override fun onStart() {
    super.onStart()
    val currentUser = auth.currentUser
    if (currentUser == null) {
        val intent = Intent(this@MainActivity, SignInActivity::class.java)
        startActivity(intent)
        finish()
    }
    else{
        updateUI(currentUser)
    }
}
```

13. Sampai tahap ini ketika aplikasi kembali di jalankan maka nama akun sudah muncul.



14. Login, Logout, Email Verification telah berhasil dibuat. Selanjutnya kita akan membuat halaman **Sign Up**.
15. Buka **activity_sign_up.xml** lalu tambahkan kode berikut ini ([link](#)). Pada **activity_sign_up.xml** terdapat **inputEmail** digunakan untuk memasukan email yang akan di daftarkan, **inputPassword** digunakan untuk memasukan password yang akan di daftarkan, **btnCreateAccount** digunakan untuk memerintahkan eksekusi create account.
16. Buka **SignUpActivity.kt** tambahkan kode berikut ini.

```

class SignUpActivity : AppCompatActivity(), View.OnClickListener {
    private lateinit var auth: FirebaseAuth
    private lateinit var binding: ActivitySignUpBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivitySignUpBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.btnCreateAccount.setOnClickListener(this)
        auth = Firebase.auth
    }

    override fun onClick(v: View) {
        when (v.id) {
            R.id.btnCreateAccount -> {
                createAccount(binding.inputEmail.text.toString(),
                    binding.inputPassword.text.toString())
            }
        }
    }
}

```

Kode di atas digunakan untuk memanggil Firebase, mengimplementasikan binding, dan memberi fungsi onclick pada **btnCreateAccount**.

17. Tambahkan fungsi **createAccount()**.

```

private fun createAccount(email: String, password: String) {
    if (!validateForm()) {
        return
    }

    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                Toast.makeText(baseContext, "Create User Success.",
                    Toast.LENGTH_SHORT).show()
                finish()
            } else {
                Toast.makeText(baseContext, "Authentication failed.",
                    Toast.LENGTH_SHORT).show()
            }
        }
}

```

18. Tambahkan fungsi **validateForm()**.

```

private fun validateForm(): Boolean {
    var valid = true

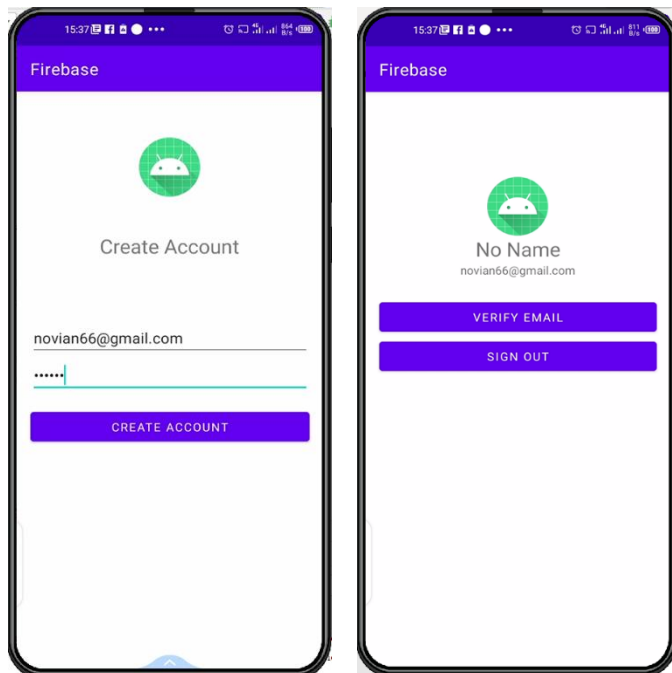
    val email = binding.inputEmail.text.toString()
    if (TextUtils.isEmpty(email)) {
        binding.inputEmail.error = "Required."
        valid = false
    } else {
        binding.inputEmail.error = null
    }

    val password = binding.inputPassword.text.toString()
    if (TextUtils.isEmpty(password)) {
        binding.inputPassword.error = "Required."
        valid = false
    } else {
        binding.inputPassword.error = null
    }

    return valid
}



```

19. Sampai tahap ini aplikasi telah mampu membuat akun baru.



Telusuri berdasarkan alamat email, nomor telepon, atau UID pengguna

Tambahkan pengguna

ID	Penyedia	Dibuat	Login	UID pengguna ↑
novian66@gmail.com		7 Jan 2021	7 Jan 2021	9ZTpmAB9p8Zg1tcHXC7yl6qcYZc2
novian@ittelkom-pwt.ac.id		7 Jan 2021	7 Jan 2021	KVTnPJPGTZMkPmrq5sFsePTtc8y2

Baris per halaman:

50

1 - 2 of 2

20. Agar ketika proses create account berhasil halaman signIn langsung tertutup tambahkan kode berikut ini pada **SignInActivity.kt**.

```
public override fun onStart() {
    super.onStart()
    val currentUser = auth.currentUser
    if (currentUser != null) {
        val intent = Intent(this@SignInActivity, MainActivity::class.java)
        startActivityForResult(intent, 1)
        finish()
    }
}
```

11.4 Auth Google

1. Buka gradle lalu tambahkan depedensi berikut ini.

```
implementation 'com.google.android.gms:play-services-auth:19.0.0'
```

2. Buka **SignInActivity.kt** lalu tambahkan **GoogleSignInClient**.

```
private lateinit var auth: FirebaseAuth
private lateinit var binding: ActivitySignInBinding
private lateinit var googleSignInClient: GoogleSignInClient
```

3. Pada onCreate tambahkan kode berikut ini.

```
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestEmail()
    .build()
googleSignInClient = GoogleSignIn.getClient(this, gso)
```

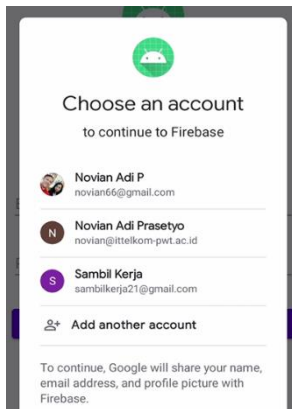
4. Pada onclick tambahkan kode berikut ini.

```
R.id.btnEmail -> {
    signIn()
}
```

5. Tambahkan fungsi **SignIn()**.

```
private fun signIn() {
    val signInIntent = googleSignInClient.signInIntent
    startActivityForResult(signInIntent, 9001)
}
```

Fungsi diatas akan membuka **SignInIntent** bawaan seperti dibawah ini.



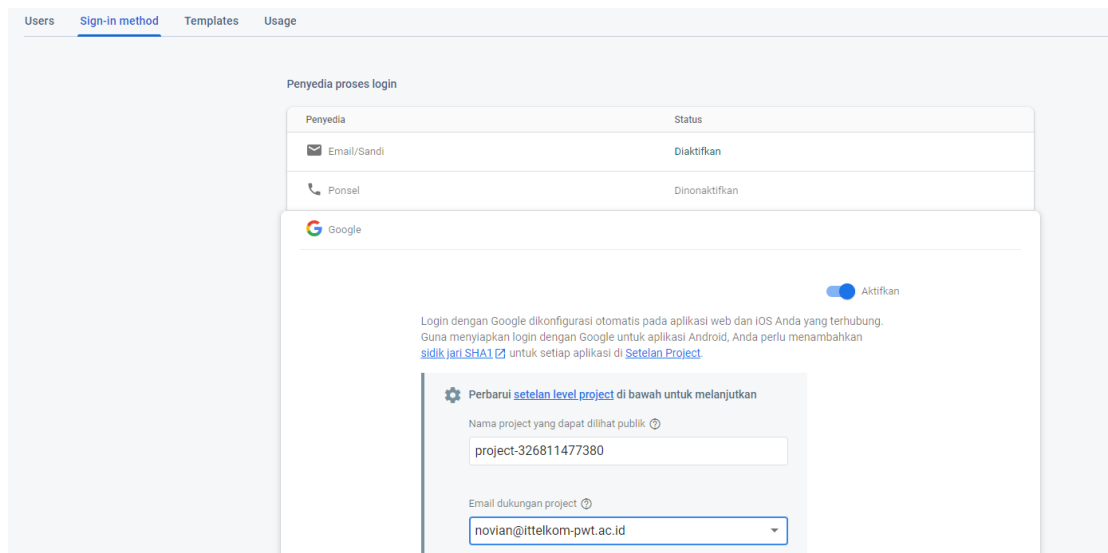
6. Sampai tahap ini ketika kita memilih salah satu account maka akan dikembalikan sebuah callback berupa request code yang dikirimkan dari **signIn()** yaitu 9001. Jika request code sesuai maka ambil **idToken** **account.idToken**. Proses ini akan dilakukan pada **onActivityResult()**.

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == 9001) {
        val task = GoogleSignIn.getSignedInAccountFromIntent(data)
        try {
            val account = task.getResult(ApiException::class.java)!!
            firebaseAuthWithGoogle(account.idToken!!)
        } catch (e: ApiException) {
            val view = binding.mainLayout
            Snackbar.make(view, "Google sign in failed.",
                Snackbar.LENGTH_SHORT).show()
        }
    }
}
```

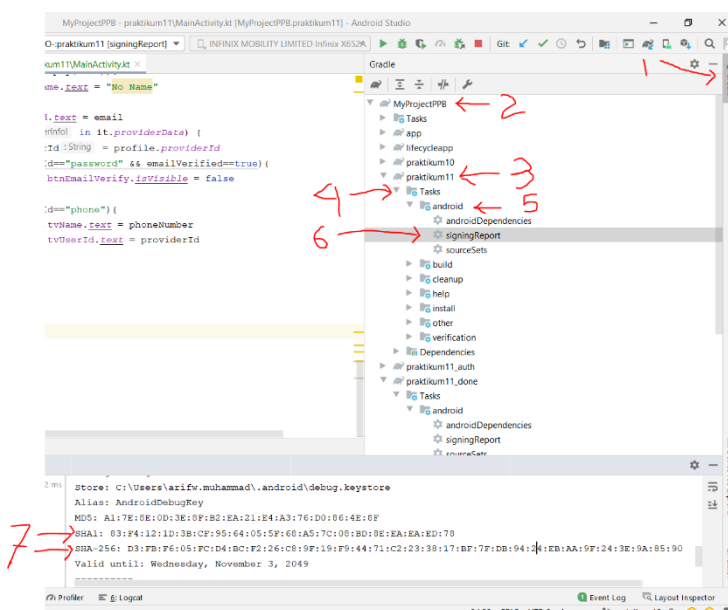
7. idToken yang didapat akan di masukan pada fungsi firebaseAuthWithGoogle().
Tambahkan kode berikut ini.

```
private fun firebaseAuthWithGoogle(idToken: String) {
    val credential = GoogleAuthProvider.getCredential(idToken, null)
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                val user = auth.currentUser
                val view = binding.mainLayout
                Snackbar.make(view, "Authentication Success.",
                    Snackbar.LENGTH_SHORT).show()
                val intent = Intent(this@SignInActivity, MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                val view = binding.mainLayout
                Snackbar.make(view, "Authentication Failed.",
                    Snackbar.LENGTH_SHORT).show()
            }
        }
}
```

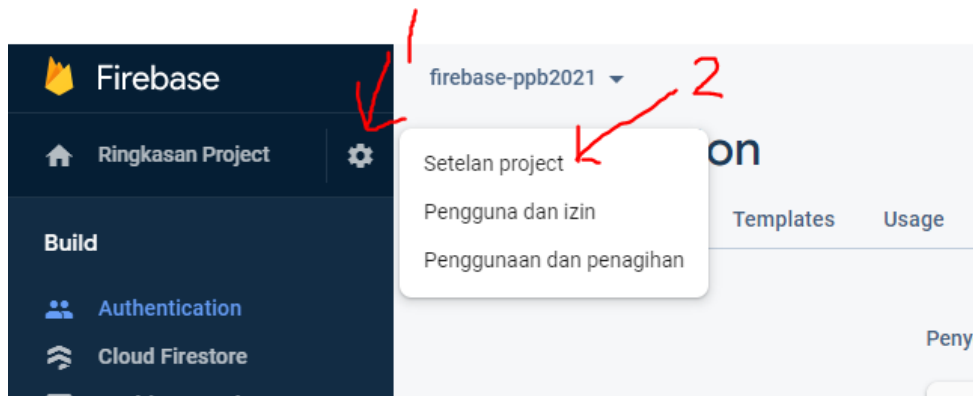
8. Sampai tahap ini koding sudah selesai, namun masih perlu terdapat tambahan konfigurasi pada firebase. Pada Sign-in method kita aktifkan google.



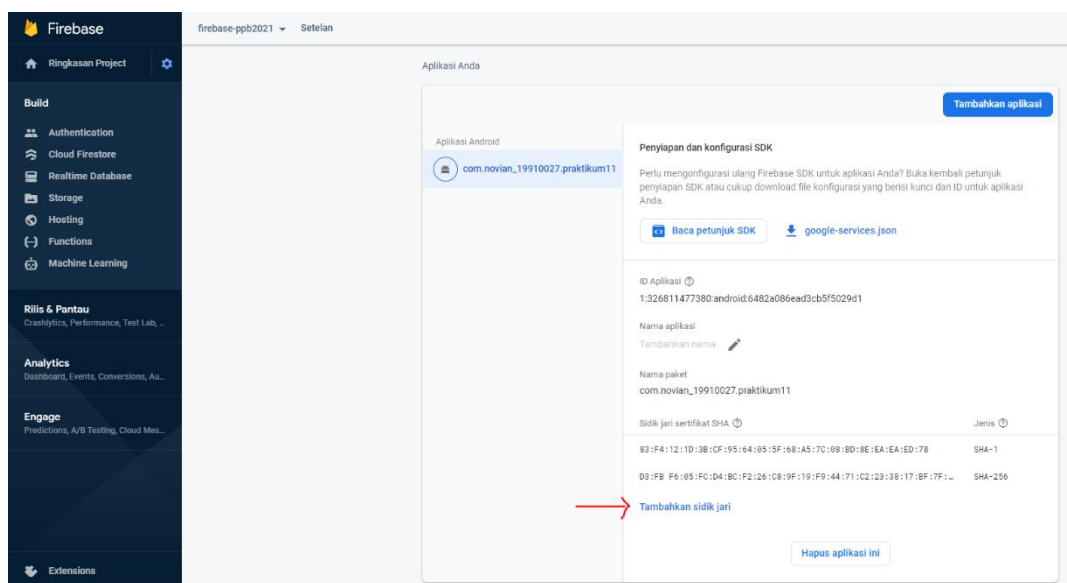
9. Kembali ke android studio, kita buka tab gradle lalu ambil SHA1 dan SHA-256.



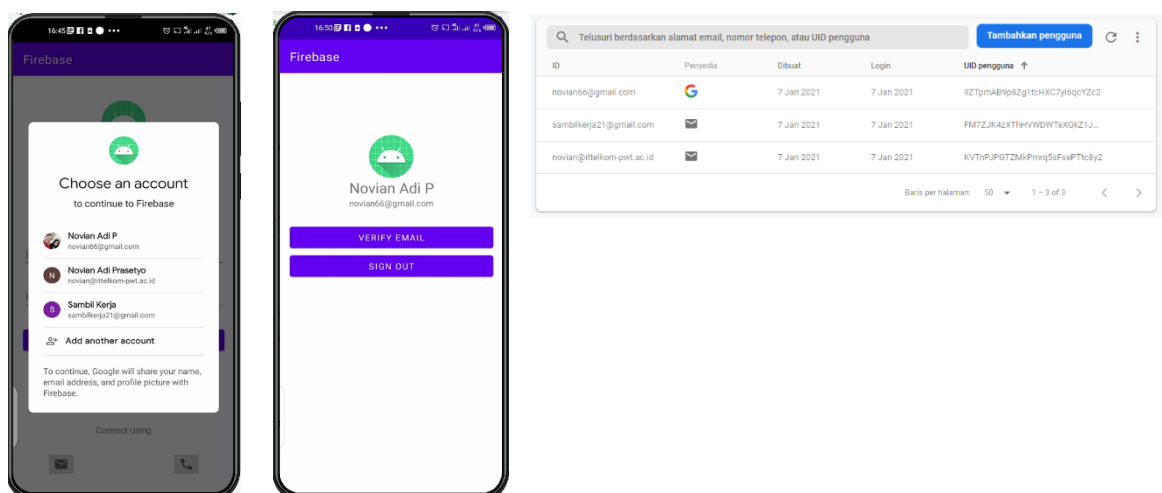
10. Kembali ke firebase, kita buka **setelan project**.



11. Tambahkan SHA1 dan SHA-256 disini.



12. Uji coba aplikasi.



13. Jika kita melakukan logout dan login kembali menggunakan google, maka kita langsung login tanpa di arahkan untuk memiliki email lagi. Untuk mengatasi hal tersebut tambahkan kode berikut pada **MainActivity.kt** di fungsi **signOut()**.

```
googleSignInClient.signOut().addOnCompleteListener(this) {  
}
```

Karena kita menggunakan **googleSignInClient** maka kita perlu memanggil kelas tersebut. Bagaimana cara memanggil kelasnya? Gunakan cara yang sama seperti pada **SignInActivity.kt**.

11.5 Tugas

1. Selesaikan modul praktikum.
2. Step tambahan :
 - a. Pada langkah **11.3.12** terdapat **currentUser.photoUrl** yang digunakan untuk mengambil foto jika login menggunakan google, tampilkan foto tersebut di **MainActivity**.
3. Pastikan tidak ada terjadi error pada aplikasi.
4. Build aplikasi menjadi .apk.
5. Upload file .apk ke lms, deadline dapat dilihat pada lms.
6. Kriteria Penilaian :
 - a. Melanjutkan project pada pertemuan kemarin, lakukan commit & push pada github (30)
 - b. Semua fitur pada module praktikum dapat dijalankan dengan baik (55)
 - c. Improvisasi (15)
 - d. Keterlambatan pengurangan nilai (20)