Institut Teknologi

# Telkom Purwokerto

2020/2021

# PEMROGRAMAN PERANGKAT BERGERAK

## Modul Praktikum Topik 12

- Konfigurasi Firestore
- CRUD Firestore
- Auth Phone Number

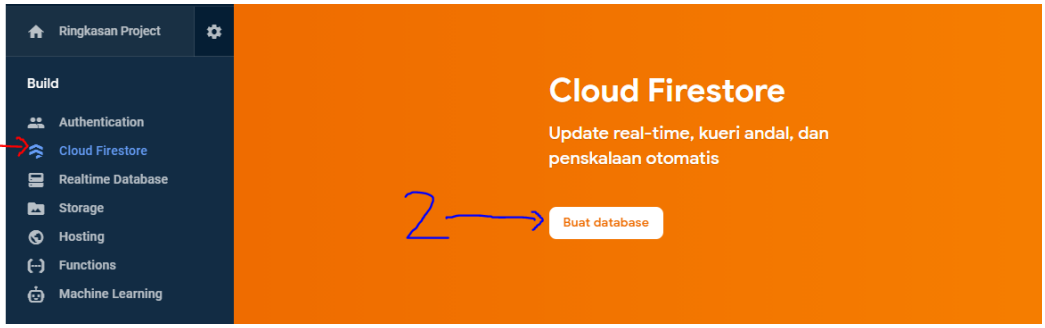
Novian Adi Prasetyo, S.Kom., M.Kom.
Condro Kartiko, S.Kom., M. T. I.

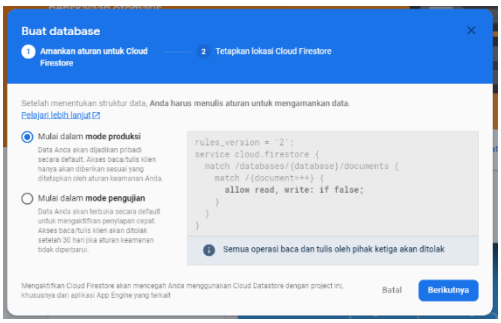
## 12.1 Persiapan

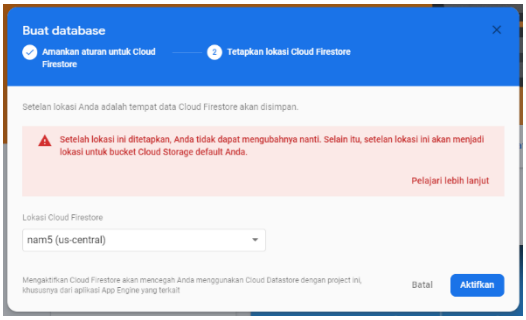
1. Android Studio
2. Device Android/Emulator Android
3. Kabel Data
4. Buka kembali module praktikum 11

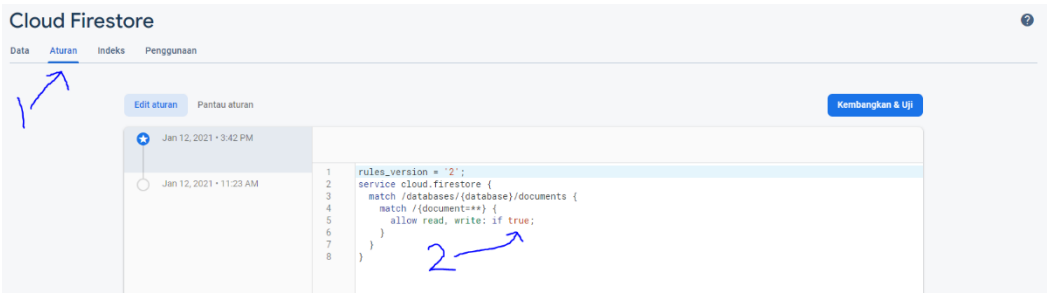
## 12.2 Konfigurasi Firestore & Halaman Quotes

1. Aktifkan cloud firestore pada https://console.firebase.google.com/.

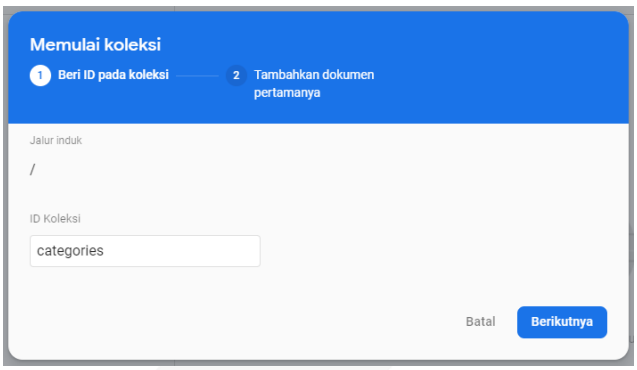


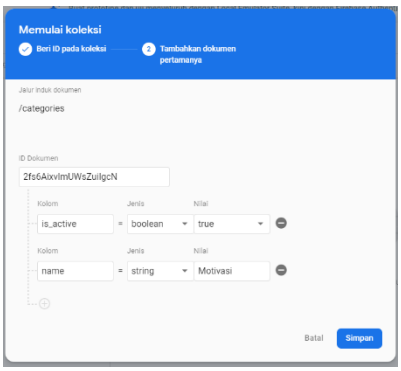




2. Buat sebuah koleksi baru dengan nama **categories**, Pada ID Dokumen buatlah menjadi otomasi, lalu tambahkan kolom **is_active** dan **name** seperti pada gambar.

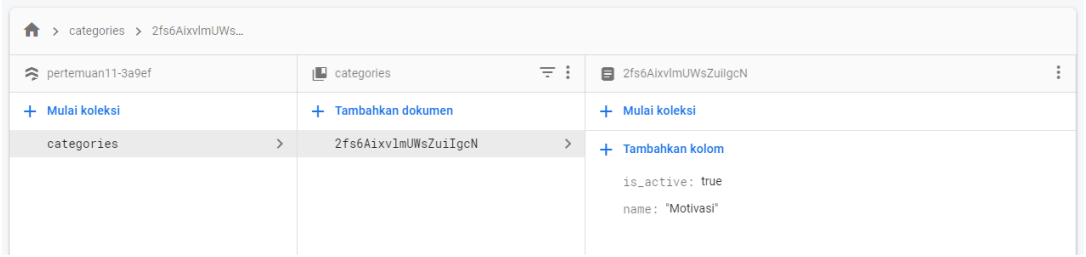





Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

3. Lanjutkan dengan membuat dokumen baru dengan menekan tombol tambah dokumen. Tambahkan untuk "Persahabatan", "Percintaan", "Keluarga", "Musik", "Film".



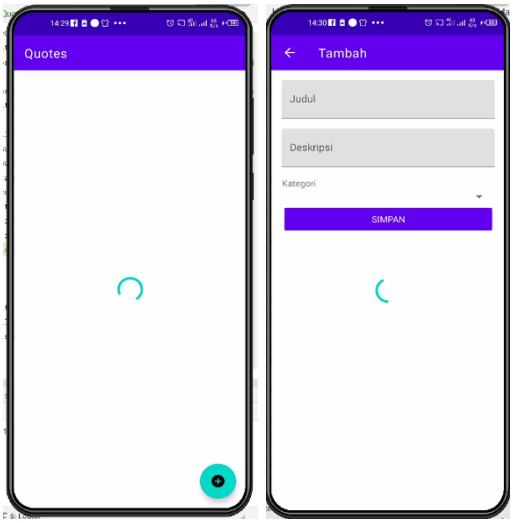

4. Sampai tahap ini cloud firestore telah berhasil diaktifkan dan kita sudah memiliki koleksi **categories**, karena kita melanjutkan project praktikum 11 maka kita sudah tidak perlu melakukan koneksi project ke firebase, kita cukup menambahkan dependecy untuk firestore. Buka gradle lalu tambahkan depedency berikut ini.

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

5. Kita akan menerapkan konsep yang sama seperti pada praktikum 10 yaitu membuat aplikasi CRUD quotes, yang membedakan adalah pada praktikum 12 ini CRUD akan tersimpan di cloud firestore. **Karena semua komponen pada praktikum 10 sudah kita pelajari sebelumnya, maka kode-kode yang terkait dengan praktikum 10 tidak ada di jelaskan lagi. Pada modul ini hanya akan berfokus membahasa kode terkait cloud firestore.**
6. Buat package baru dengan nama **data**, didalamnya buat class baru dengan nama **Quote**, tambahkan kode berikut ini **link**. Kode tersebut berisi sama persis dengan **Quote** pada praktikum 10.
7. Buat package baru dengan nama **adapater**, didalamnya buat class baru dengan nama **QuoteAdapter,** tambahkan kode berikut ini **link**. Kode tersebut merupakan ringkasan dari **QuoteAdapter** pada praktikum 10 dengan menghilangkan beberapa kode yang terkait dengan SQLite.
8. Buat object baru dengan nama **helper,** tambahkan kode berikut ini **link**. Kode tersebut merupakan ringkasan dari **helper** pada praktikum 10 dengan menghilangkan beberapa kode yang terkait dengan SQLite.
9. Buat activity baru dengan nama **DashboardQuoteActivity.** Lalu tambahkan kode berikut ini **link**. Kode tersebut adalah ringkasan dari MainActivity pada praktikum 10 dengan menghapus beberapa kode terkait SQLite.
10. Pada **activity_dashboard_quote.xml** tambahkan kode berikut ini **link.** Kode tersebut berisi sama persis dengan **main_activity.xml** pada praktikum 10.
11. Buat activity baru dengan nama **QuoteAddUpdateActivity.** Lalu tambahkan kode berikut ini **link**. Kode tersebut adalah ringkasan dari **QuoteAddUpdateActivity** pada praktikum 10 dengan menghapus beberapa kode terkait SQLite.
12. Pada **activity_quote_add_update.xml** tambahkan kode berikut ini **link.** Kode tersebut berisi sama persis dengan **activity_quote_add_update.xml** pada praktikum 10.
13. Buat layout baru dengan nama **item_quote.xml** tambahkan kode berikut ini **link.** Kode tersebut berisi sama persis dengan **item_quote.xml** pada praktikum 10.


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

14. Pada **MainActivity** buatlah sebuah tombol yang dapat memberikan akses untuk membuka **DashboardQuoteActivity**. Jangan lupa tambahkan **setOnClickListener** pada tombol tersebut agar dapat memiliki aksi.
15. Sampai tahap ini jika diuji halaman quotes sudah dapat dijalankan namun dengan fungsi CRUD yang belum dapat bekerja.




## 12.3 CREATE

1. Buka **QuoteAddUpdateActivity** tambahkan kode berikut ini menginisasikan firestore.

```
private lateinit var auth: FirebaseAuth
private lateinit var firestore: FirebaseFirestore
```

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityQuoteAddUpdateBinding.inflate(layoutInflater)
    setContentView(binding.root)
    firestore = Firebase.firestore
    auth = Firebase.auth
    categoriesSpinnerArray = getCategories()
    quote = intent.getParcelableExtra(EXTRA_QUOTE)
```

Disini kita masih menggunakan FirebaseAuth untuk mengambil UID yang nantinya akan dimasukan kedalam setiap quote yang dibuat.

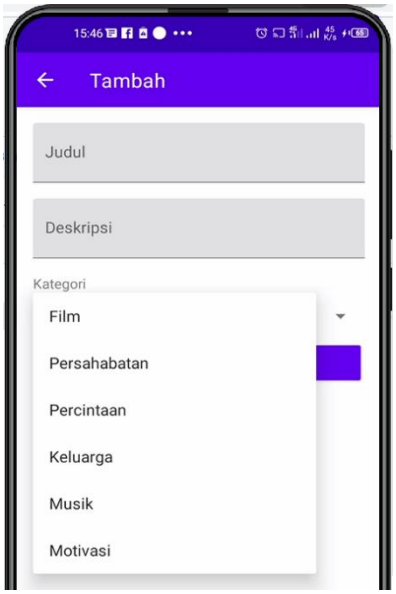
2. Pada **getCategories()** tambahkan kode berikut ini.

```
private fun getCategories(): ArrayList<String> {
    progressbar.visibility = View.VISIBLE
    firestore.collection("categories")
        .whereEqualTo("is_active", true)
        .get()
        .addOnSuccessListener { documents ->
            var selection = 0;
            for (document in documents) {
                val name = document.get("name").toString()
                quote?.let {
                    if(name==it.category){
                        categorySelection = selection
                    }
                }
                categoriesSpinnerArray.add(name)
                selection++
            }
            setCategories(categoriesSpinnerArray)
        }
        .addOnFailureListener { exception ->
            Toast.makeText(this@QuoteAddUpdateActivity, "Categories cannot be
retrieved ", Toast.LENGTH_SHORT).show()
        }
    return categoriesSpinnerArray
}
```


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

Pada kode diatas kita mulai mengambil data pada koleksi **categories** dengan perintah **firestore.collection("categories")**, lalu kita menambahkan sebuah filter dengan **whereEqualTo("is_active", true).get()** agar data yang diambil berupa data dengan **is_active = true**. Jika data berhasil diambil selanjutnya **addOnSuccessListener** akan bekerja dan melakukan iterasi pada data yang berhasil diambil, hasil iterasi tersebut digunakan untuk memasukan data ke spinner category.

3. Sampai tahap ini jika aplikasi diuji maka kolom kategori sudah dapat menampilkan data sesuai koleksi categories yang ada di firestore kita**.**




4. Selanjutnya pada onClick tambahkan kode berikut.

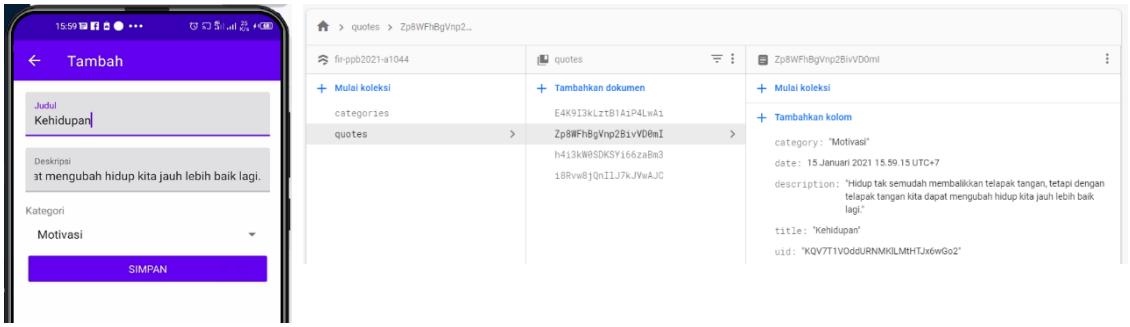
```
override fun onClick(view: View) {
    if (view.id == R.id.btn_submit) {
        val title = binding.edtTitle.text.toString().trim()
        val description = binding.edtDescription.text.toString().trim()
        if (title.isEmpty()) {
            binding.edtTitle.error = "Field can not be blank"
            return
        }
        if (isEdit) {

        } else {
            val currentUser = auth.currentUser
            val user = hashMapOf(
                "uid" to currentUser?.uid,
                "title" to title,
                "description" to description,
                "category" to categoryName,
                "date" to FieldValue.serverTimestamp()
            )
            firestore.collection("quotes")
                .add(user)
                .addOnSuccessListener { documentReference ->
                    Toast.makeText(this@QuoteAddUpdateActivity,
"DocumentSnapshot added with ID: ${documentReference.id}",
Toast.LENGTH_SHORT).show()
                    setResult(RESULT_ADD, intent)
                    finish()
                }
                .addOnFailureListener { e ->
                    Toast.makeText(this@QuoteAddUpdateActivity, "Error adding
document", Toast.LENGTH_SHORT).show()
                }
        }
    }
}
```



Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

Kode diatas akan menjalankan proses simpan data ke firestore ketika tombol simpan ditekan, pada firestore kita tidak perlu membuat koleksi terlebih dahulu agar data dapat simpan, kita dapat langsung memanggil **firestore.collection("quotes").add(user)** dengan begitu jika koleksi belum ada maka akan dibuat koleksi dengan nama tersebut. Variabel **user** digunakan untuk memetakan data yang akan disimpan pada koleksi **quotes.** Dengan begitu kita dapat melihat terdapat 5 kolom yang akan terbentuk yaitu uid, title, description, category, dan date.

5. Lakukan uji coba menyimpan data, hasilnya seperti gambar dibawah ini.




## 12.4 READ

1. Buka **DashboardQuoteActivity** tambahkan kode berikut ini menginisasikan firestore.

```
private lateinit var auth: FirebaseAuth
private lateinit var firestore: FirebaseFirestore
```

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityDashboardQuoteBinding.inflate(layoutInflater)
    setContentView(binding.root)
    firestore = Firebase.firestore
    auth = Firebase.auth
    supportActionBar?.title = "Quotes"
    binding.rvQuotes.layoutManager = LinearLayoutManager(this)
```

Disini kita masih menggunakan FirebaseAuth untuk mengambil UID yang nantinya akan digunakan untuk mengseleksi quote yang akan ditampilkan.

2. Pada **loadQuotes()** tambahkan kode berikut.

```
private fun loadQuotes() {
    GlobalScope.launch(Dispatchers.Main) {
        progressbar.visibility = View.VISIBLE
        val quotesList = ArrayList<Quote>()
        val currentUser = auth.currentUser
        firestore.collection("quotes")
            .whereEqualTo("uid", currentUser?.uid)
            .get()
            .addOnSuccessListener { result ->
                progressbar.visibility = View.INVISIBLE
                for (document in result) {
                    val id = document.id
                    val title = document.get("title").toString()
                    val description = document.get("description").toString()
                    val category = document.get("category").toString()
                    val date = document.get("date") as com.google.firebase.Timestamp
                    quotesList.add(Quote(id, title, description, category, date))
                }
                if (quotesList.size > 0) {
                    binding.rvQuotes.adapter = adapter
                    adapter.listQuotes = quotesList
                } else {
                    adapter.listQuotes = ArrayList()
                    showSnackbarMessage("Tidak ada data saat ini")
                }
            }
            .addOnFailureListener { exception ->
                progressbar.visibility = View.INVISIBLE
                Toast.makeText(
                    this@DashboardQuoteActivity, "Error adding document",Toast.LENGTH_SHORT
                ).show()
            }
    }
}
```


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

Kode diatas akan memanggil data dari koleksi **quotes** dengan perintah **firestore.collection("quotes").** Sedangkan **whereEqualTo("uid", currentUser?.uid).get()** digunakan untuk seleksi data yaitu mengambil data quote berdasarkan uid yang sedang login saat ini. Ketika data berhasil didapatkan maka akan menjalankan **addOnSuccessListener** yang didalamnya akan melakukan iterasi data lalu memasukan data tersebut kesebuah array list.

3. Lakukan uji coba membuka halaman **DashboardQuoteActivity**, hasilnya seperti gambar dibawah ini.


## 12.5 UPDATE

1. Buka **QuoteAddUpdateActivity** tambahkan kode berikut ini.

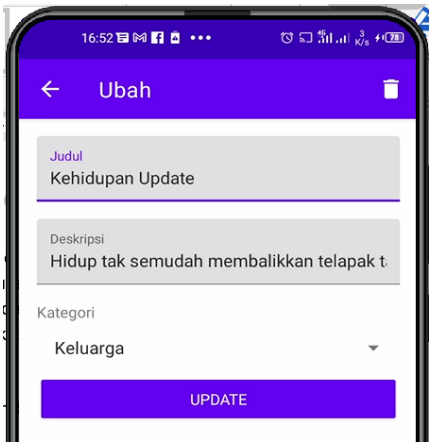
```
override fun onClick(view: View) {
    if (view.id == R.id.btn_submit) {
        val title = binding.edtTitle.text.toString().trim()
        val description = binding.edtDescription.text.toString().trim()
        if (title.isEmpty()) {
            binding.edtTitle.error = "Field can not be blank"
            return
        }
        if (isEdit) {
            val currentUser = auth.currentUser
            val user = hashMapOf(
                "uid" to currentUser?.uid,
                "title" to title,
                "description" to description,
                "category" to categoryName,
                "date" to FieldValue.serverTimestamp()
            )

            firestore.collection("quotes").document(quote?.id.toString())
                .set(user)
                .addOnSuccessListener {
                    setResult(RESULT_UPDATE, intent)
                    finish()
                }
                .addOnFailureListener { e ->
                    Toast.makeText(this@QuoteAddUpdateActivity, "Gagal
mengupdate data", Toast.LENGTH_SHORT).show()
                }
        } else {
```

Kode diatas akan dijalankan ketika tombol edit di tekan, isi dari kode tersebut adalah mengirim data yang sebelumnya telah dipetakan pada variabel user, lalu variabel user tersebut dikirimkan ke firestore dengan perintah **firestore.collection("quotes").** Kode **.document(quote?.id.toString())** digunakan untuk mengseleksi dokumen yang


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

akan diupdate sesuai dengan dokumen yang dipilih. Kode **.set(user)** digunakan untuk menjalankan proses update dengan data **user.** Jika update berhasil maka **addOnSuccessListener** dijalankan, didalamnya akan dieksekusi untuk menutup halaman **QuoteAddUpdateActivity.**

2. Lakukan uji coba mengubah salah satu data, hasilnya seperti gambar dibawah ini.





## 12.6 DELETE

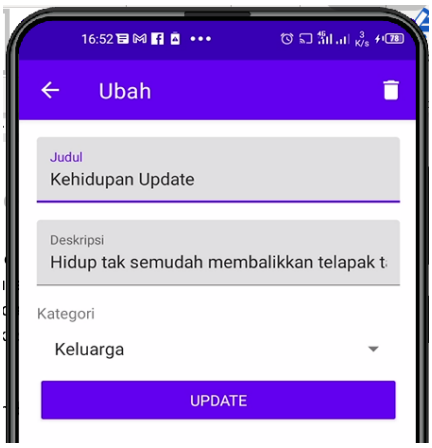
1. Buka **QuoteAddUpdateActivity** tambahkan kode berikut ini pada showAlertDialog(type: Int).

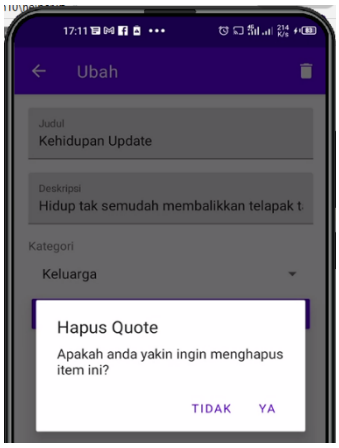
```
val alertDialogBuilder = AlertDialog.Builder(this)
alertDialogBuilder.setTitle(dialogTitle)
alertDialogBuilder
    .setMessage(dialogMessage)
    .setCancelable(false)
    .setPositiveButton("Ya") { _, _ ->
        if (isDialogClose) {
            finish()
        } else {
            firestore.collection("quotes").document(quote?.id.toString())
                .delete()
                .addOnSuccessListener {
                    Log.d("delete", "DocumentSnapshot successfully
deleted!"+quote?.id.toString())
                                        val intent = Intent()
                intent.putExtra(EXTRA_POSITION, position)
                setResult(RESULT_DELETE, intent)
                finish()
                }
                .addOnFailureListener { e ->
                    Log.w("a", "Error deleting document", e)
                    Toast.makeText(this@QuoteAddUpdateActivity, "Gagal menghapus
data", Toast.LENGTH_SHORT).show()
                }
        }
```

Kode diatas akan dijalankan ketika tombol hapus ditekan, kode **firestore.collection("quotes")** digunakan untuk menentukan koleksi yang akan dihapus. Kode **document(quote?.id.toString()).delete()** digunakan untuk mengseleksi dokumen yang akan dihapus.


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

2. Lakukan uji coba menghapus salah satu data, hasilnya seperti gambar dibawah ini.




## 12.7 AUTH PHONE NUMBER

1. Buat activity baru dengan nama **PhoneAuthActivity**, tambahkan kode berikut ini.

```
private lateinit var auth: FirebaseAuth
private lateinit var binding: ActivityPhoneAuthBinding
private var verificationInProgress = false
private var storedVerificationId: String? = ""
private var prefixPhoneNumber: String? = "+62"
private lateinit var resendToken: PhoneAuthProvider.ForceResendingToken
private lateinit var callbacks:
PhoneAuthProvider.OnVerificationStateChangedCallbacks
```

2. Pada **activity_phone_auth.xml** tambahkan kode berikut ini link.
3. Pada onCreate ubah menjadi seperti dibawah ini.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityPhoneAuthBinding.inflate(layoutInflater)
    setContentView(binding.root)

    binding.btnVerify.setOnClickListener(this)
    binding.btnContinue.setOnClickListener(this)
    binding.btnResend.setOnClickListener(this)
    auth = Firebase.auth
}
```

4. Tambahkan onClick seperti dibawah ini.

```
override fun onClick(view: View) {
    when (view.id) {
        R.id.btnVerify -> {
            val phoneNumber = binding.inputNumber.text.toString()
            if (TextUtils.isEmpty(phoneNumber)) {
                binding.inputNumber.error = "Invalid phone number."
                return
            }

startPhoneNumberVerification(prefixPhoneNumber+binding.inputNumber.text.toString
())
        }
        R.id.btnContinue -> {
            val code = binding.inputVerifyCode.text.toString()
            if (TextUtils.isEmpty(code)) {
                binding.inputVerifyCode.error = "Cannot be empty."
                return
            }
            verifyPhoneNumberWithCode(storedVerificationId, code)
        }
        R.id.btnResend ->
resendVerificationCode(prefixPhoneNumber+binding.inputNumber.text.toString(),
resendToken)
    }
}
```


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

Kode diatas akan memberikan aksi kepada tombol **btnVerify** untuk memulai verifikasi nomor telp. Tombol **btnContinue** digunakan untuk memverifikasi kode yang diterima. Tombol **btnResend** digunakan untuk permintaan ulang pengiriman verifikasi nomor telp.

5. Selanjutnya kita akan membuat fungsi **startPhoneNumberVerification()** sebagai fungsi yang akan bekerja jika btnVerify ditekan.

```
private fun startPhoneNumberVerification(phoneNumber: String) {
    val options = PhoneAuthOptions.newBuilder(auth)
        .setPhoneNumber(phoneNumber)
        .setTimeout(60L, TimeUnit.SECONDS)
        .setActivity(this)
        .setCallbacks(callbacks)
        .build()
    PhoneAuthProvider.verifyPhoneNumber(options)
    verificationInProgress = true
}
```

6. Setelah fungsi diatas berjalan, selanjutnya akan dipanggil sebuah callback **PhoneAuthProvider**. Tambahkan kode berikut ini pada **onCreate** untuk menangani callback tersebut.

```
callbacks = object : PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
    override fun onVerificationCompleted(credential: PhoneAuthCredential) {
        verificationInProgress = false
        signInWithPhoneAuthCredential(credential)
    }
    override fun onVerificationFailed(e: FirebaseException) {
        verificationInProgress = false
        if (e is FirebaseAuthInvalidCredentialsException) {
            binding.inputNumber.error = "Invalid phone number."
        } else if (e is FirebaseTooManyRequestsException) {
            Snackbar.make(findViewById(android.R.id.content), "Quota exceeded.",
                Snackbar.LENGTH_SHORT).show()
        }
    }
    override fun onCodeSent(
        verificationId: String,
        token: PhoneAuthProvider.ForceResendingToken
    ) {
        storedVerificationId = verificationId
        resendToken = token
    }
}
```

Pada callback diatas kita sudah dapat menyimpulkan bahwa ketika verifikasi berhasil maka **onVerificationCompleted** akan bekerja, ketika kode sudah dikirimkan maka **onCodeSent** akan bekerja, ketika verifikasi gagal maka **onVerificationFailed** akan bekerja.

7. Saat **onVerificationCompleted** bekerja, didalamnya akan memanggil sebuah fungsi **signInWithPhoneAuthCredential().** Sekarang tambahkan fungsi tersebut seperti dibawah ini.

```
private fun signInWithPhoneAuthCredential(credential: PhoneAuthCredential) {
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                finish()
            } else {
                if (task.exception is FirebaseAuthInvalidCredentialsException) {
                    binding.inputVerifyCode.error = "Invalid code."
                }
            }
        }
}
```



Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

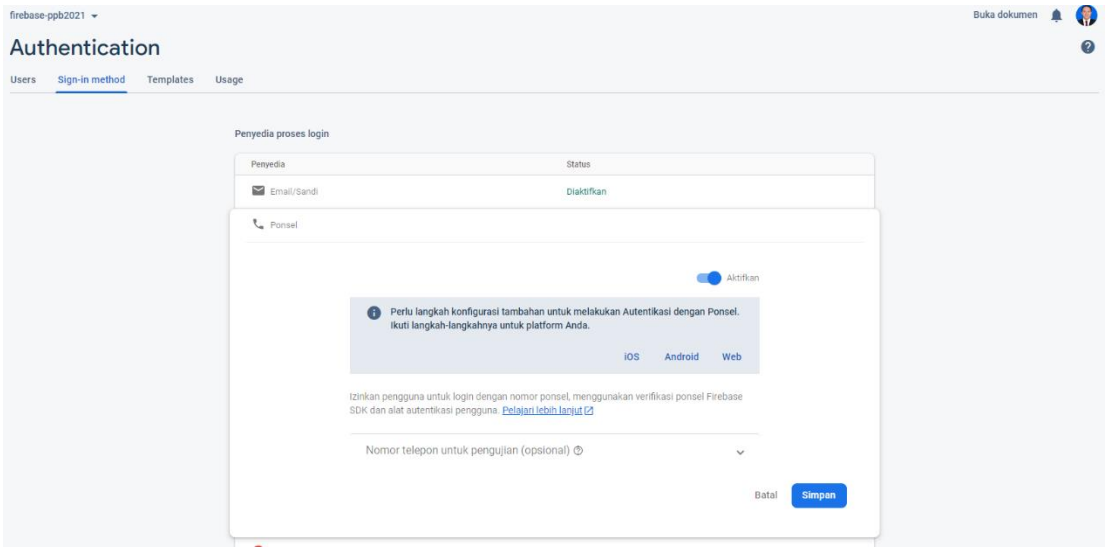
8. Kita lihat pada langkah 12.7.4 terdapat **btnContinue** yang akan menjalankan fungsi **verifyPhoneNumberWithCode(),** mari kita buat fungsi berikut.

```
private fun verifyPhoneNumberWithCode(verificationId: String?, code: String) {
    val credential = PhoneAuthProvider.getCredential(verificationId!!, code)
    signInWithPhoneAuthCredential(credential)
}
```

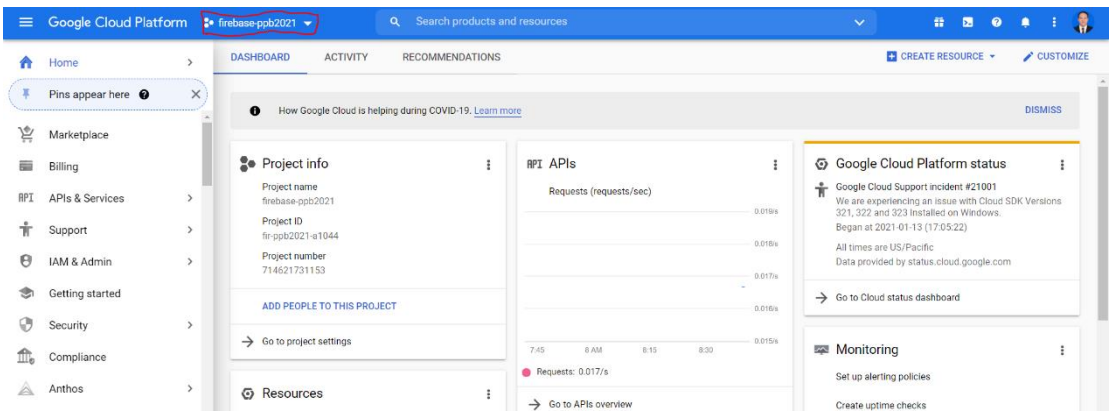
9. Kita lihat pada langkah 12.7.4 terdapat **btnResend** yang akan menjalankan fungsi **resendVerificationCode(),** mari kita buat fungsi berikut.

```
private fun resendVerificationCode(
    phoneNumber: String,
    token: PhoneAuthProvider.ForceResendingToken?
) {
    val optionsBuilder = PhoneAuthOptions.newBuilder(auth)
        .setPhoneNumber(phoneNumber)
        .setTimeout(60L, TimeUnit.SECONDS)
        .setActivity(this)
        .setCallbacks(callbacks)
    if (token != null) {
        optionsBuilder.setForceResendingToken(token)
    }
    PhoneAuthProvider.verifyPhoneNumber(optionsBuilder.build())
}
```

10. Sampai tahap ini semua kode sudah kita tambahkan, sebelum menguji kita perlu mengaktifkan fitur ponsel pada auth firebase.




11. Setelah itu kita perlu mengaktifkan Android Device Verification pada **console.cloud.google.com.** Setelah dibuka akan tampil halaman seperti dibawah ini.

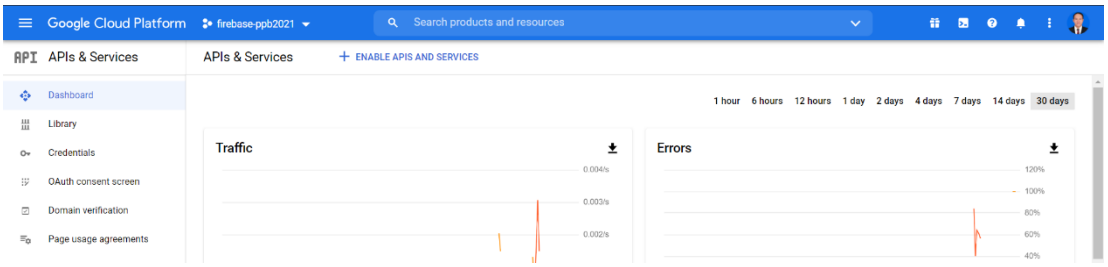



Pastikan project firebase sudah terpilih sesuai yang akan digunakan.

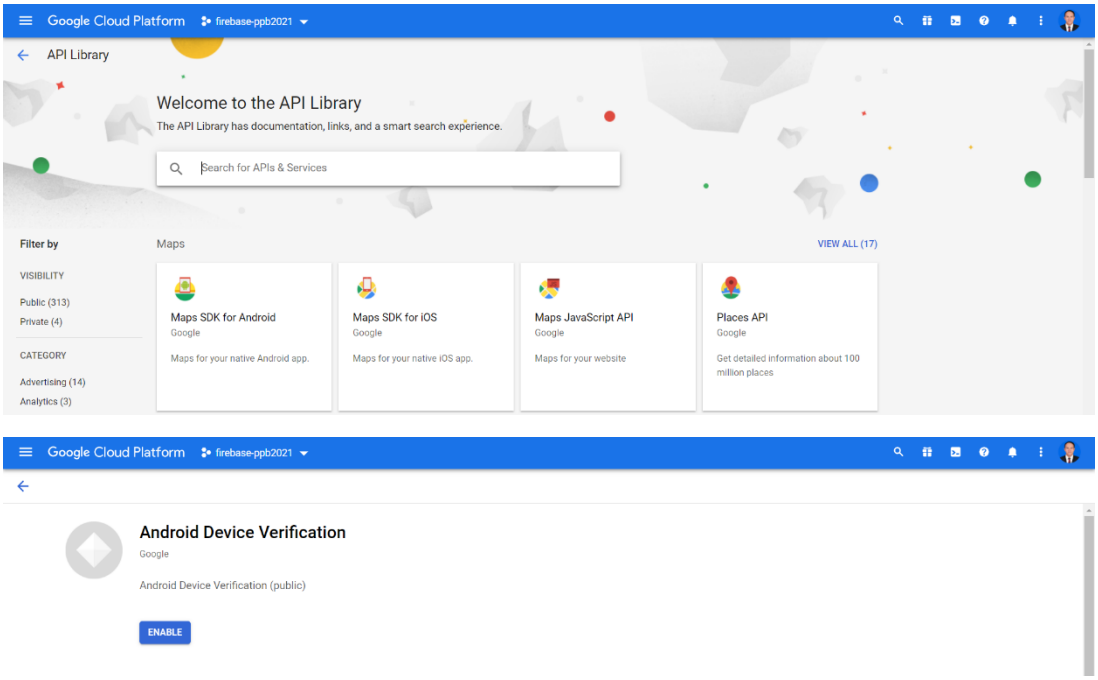


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

12. Pilih menu APIs & Services lalu klik tombol **ENABLE APIS AND SERVICES.**



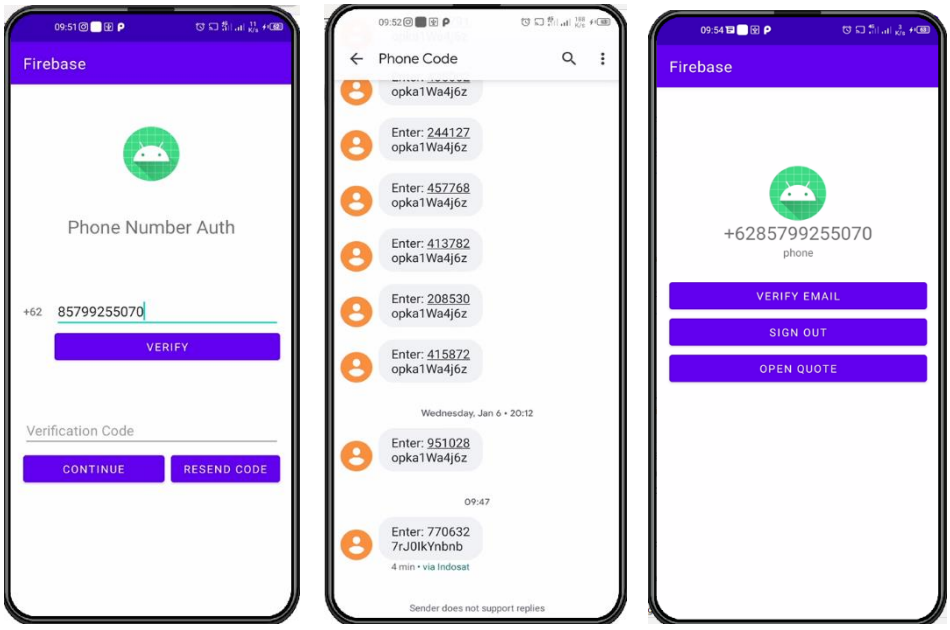

13. Pada kolom search ketilah Android Device Verification lalu klik tombol **ENABLE**.







14. Setelah itu mari kita lanjutkan login menggunakan nomor handphone.




Beberapa ponsel saat verifikasi berhasil ada yang langsung otomatis login tanpa harus mengisi kode verifikasi, namun ada juga yang harus mengisi kode verifikasi terlebih dahulu.



Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282

## 12.8 TUGAS

1. Selesaikan modul praktikum.
2. Pastikan tidak ada terjadi error pada aplikasi.
3. Build aplikasi menjadi .apk.
4. Upload file .apk ke lms, deadline dapat dilihat pada lms.
5. 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)


Novian Adi Prasetyo, S.Kom., M.Kom. | novian@ittelkom-pwt.ac.id | 085799255070
Condro Kartiko, S.Kom., M. T. I. | condro.kartiko@ittelkom-pwt.ac.id |08562637282