

Firestore

1. AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="polinema.mi.appx0d">

    <uses-permission
        android:name="android.permission.INTERNET"/>

    <application
        android:fullBackupContent="true"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity" android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

2. build.gradle (Project)

```
// Top-level build file where you can add configuration options common to all sub-
// projects/modules.

buildscript {
    ext.kotlin_version = '1.8.10'
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath 'com.google.gms:google-services:4.3.15'
        classpath 'com.android.tools.build:gradle:8.1.1'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
```

```

    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}

```

3. build.gradle (module)

```

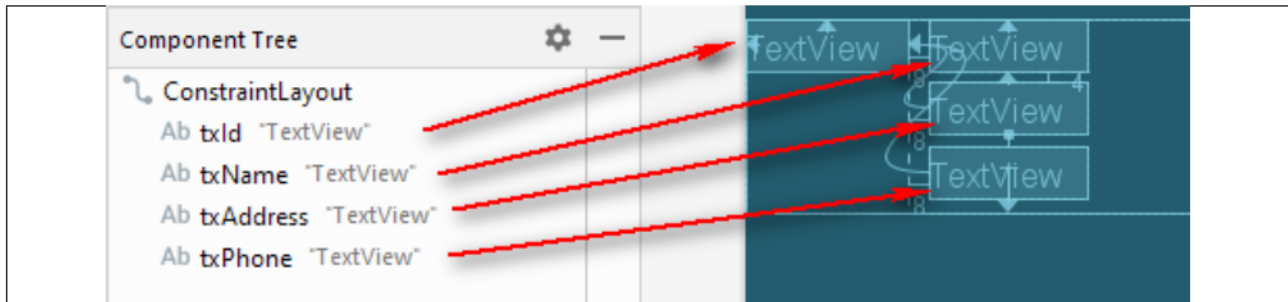
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 33
    buildToolsVersion '30.0.3'
    defaultConfig {
        applicationId "polinema.mi.appx0d"
        minSdkVersion 22
        targetSdkVersion 33
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner 'androidx.test.runner.AndroidJUnitRunner'
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
                'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
    buildFeatures{
        viewBinding true
    }
    namespace 'polinema.mi.appx0d'
}

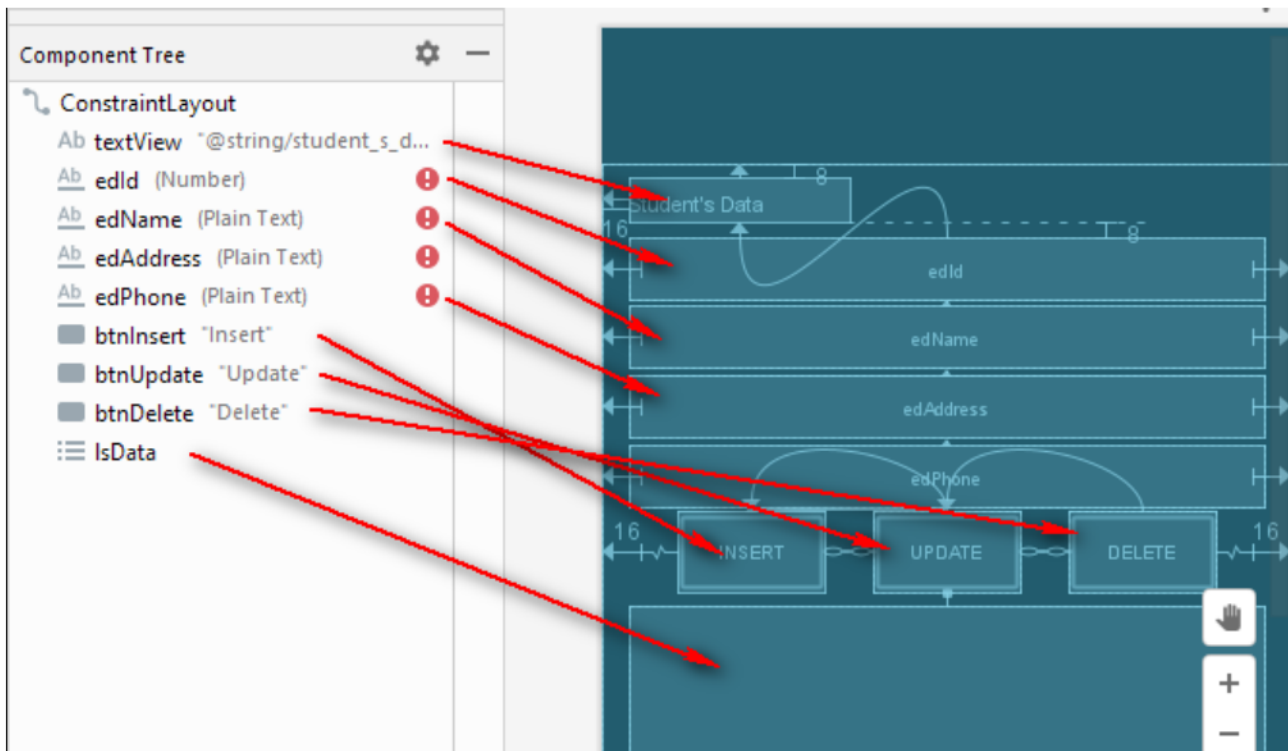
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation"org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'androidx.core:core-ktx:1.10.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test:runner:1.5.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.google.firebase:firebase-firestore:24.7.1'
    implementation 'com.google.firebase:firebase-core:21.1.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.android.material:material:1.9.0'
}

```

4. row_data.xml



5. activity_main.xml



6. MainActivity.kt (class)

```
package polinema.mi.appx0d

import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.AdapterView
import android.widget.SimpleAdapter
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.firestore.FirebaseFirestore
import polinema.mi.appx0d.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity(), View.OnClickListener {

    val COLLECTION = "students"
    val F_ID = "id"
    val F_NAME = "name"
    val F_ADDRESS = "address"
    val F_PHONE = "phone"
```

```

var docId = "" //update delete
lateinit var db : FirebaseFirestore
lateinit var alStudent : ArrayList<HashMap<String,Any>>
lateinit var adapter: SimpleAdapter
lateinit var b: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    b = ActivityMainBinding.inflate(layoutInflater)
    setContentView(b.root)

    alStudent = ArrayList()
    b.btnInsert.setOnClickListener(this)
    b.btnUpdate.setOnClickListener(this)
    b.btnDelete.setOnClickListener(this)
    b.lsData.setOnItemClickListener(itemClick)
}

override fun onStart() {
    super.onStart()
    db = FirebaseFirestore.getInstance()
    db.collection(COLLECTION).addSnapshotListener { querySnapshot, e ->
        if(e != null) Log.d("fireStore", e.localizedMessage)
        showData()
    }
}

override fun onClick(v: View?) {
    when(v?.id){
        R.id.btnInsert ->{
            val hm = HashMap<String, Any>()
            hm.set(F_ID,b.edId.text.toString())
            hm.set(F_NAME,b.edName.text.toString())
            hm.set(F_ADDRESS,b.edAddress.text.toString())
            hm.set(F_PHONE,b.edPhone.text.toString())

            db.collection(COLLECTION).document(b.edId.text.toString()).set(hm).
                addOnSuccessListener {
                    Toast.makeText(this,"Data successfully added",Toast.LENGTH_SHORT)
                        .show()
                    b.edId.setText(""); b.edName.setText(""); b.edPhone.setText("")
                    b.edAddress.setText("")
                }.addOnFailureListener { e ->
                    Toast.makeText(this,
                        "Data unsuccessfully added : ${e.message}",Toast.LENGTH_SHORT)
                        .show()
                }
        }

        R.id.btnUpdate ->{
            val hm = HashMap<String, Any>()
            hm.set(F_ID,docId)
            hm.set(F_NAME,b.edName.text.toString())
            hm.set(F_ADDRESS,b.edAddress.text.toString())
            hm.set(F_PHONE,b.edPhone.text.toString())
            db.collection(COLLECTION).document(docId).update(hm)
                .addOnSuccessListener {
                    Toast.makeText(this,"Data successfully updated",Toast.LENGTH_SHORT)
                        .show()
                }
        }
    }
}

```

```

        b.edId.setText(""); b.edName.setText(""); b.edPhone.setText("")
        b.edAddress.setText("")
    }.addOnFailureListener { e ->
        Toast.makeText(this,
            "Data unsuccessfully updated : ${e.message}", Toast.LENGTH_SHORT)
            .show()
    }
}

R.id.btnDelete -> { //contoh kita cari id = 123
    db.collection(COLLECTION).whereEqualTo(F_ID, docId).get()
    .addOnSuccessListener {
        results ->
        for(doc in results){
            db.collection(COLLECTION).document(doc.id).delete()
            .addOnSuccessListener {
                Toast.makeText(this, "Data successfully deleted",
                    Toast.LENGTH_SHORT).show()
            }.addOnFailureListener { e ->
                Toast.makeText(this, "Data unsuccessfully deleted ${e.message}",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }.addOnFailureListener { e ->
        Toast.makeText(this, "Can't get data references ${e.message}",
            Toast.LENGTH_SHORT).show()
    }
}
}
}

val itemClick = AdapterView.OnItemClickListener { parent, view, position, id ->
    val hm = alStudent.get(position)
    docId = hm.get(F_ID).toString()
    b.edId.setText(docId)
    b.edName.setText(hm.get(F_NAME).toString())
    b.edAddress.setText(hm.get(F_ADDRESS).toString())
    b.edPhone.setText(hm.get(F_PHONE).toString())
}

fun showData(){
    db.collection(COLLECTION).get().addOnSuccessListener { result ->
        alStudent.clear()
        for(doc in result){
            val hm = HashMap<String, Any>()
            hm.set(F_ID, doc.get(F_ID).toString())
            hm.set(F_NAME, doc.get(F_NAME).toString())
            hm.set(F_ADDRESS, doc.get(F_ADDRESS).toString())
            hm.set(F_PHONE, doc.get(F_PHONE).toString())
            alStudent.add(hm)
        }
        adapter = SimpleAdapter(this, alStudent, R.layout.row_data,
            arrayOf(F_ID, F_NAME, F_ADDRESS, F_PHONE),
            intArrayOf(R.id.txId, R.id.txName, R.id.txAddress, R.id.txPhone))
        b.lsData.adapter = adapter
    }
}
}

```