MODUL 10
Coroutine and Room

CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat menggunakan menu dan Dialog.

KEBUTUHAN ALAT/BAHAN/SOFTWARE

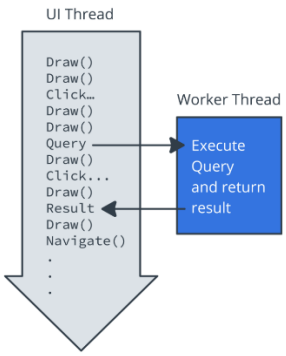
1. Android Studio 3.4.
2. Handphone Android versi 7.0 (Nougat)
3. Kabel data USB.
4. Driver ADB.

DASAR TEORI

https://codelabs.developers.google.com/codelabs/kotlin-android-training-coroutines-and-room/#0

Coroutines

Di Kotlin, coroutine adalah cara untuk menangani task yang sudah berjalan lama secara elegan dan efisien. Kotlin coroutine memungkinkan kita mengonversi kode berbasis panggilan balik ke kode sekuensial. Kode yang ditulis secara berurutan biasanya lebih mudah dibaca dan bahkan dapat menggunakan fitur bahasa seperti pengecualian. Pada akhirnya, coroutine dan callback melakukan hal yang sama: keduanya menunggu hingga hasilnya tersedia dari task yang sudah berjalan lama dan melanjutkan eksekusi.



Coroutine memiliki sifat-sifat berikut:

- Coroutine asynchronous dan non-blocking.

- Coroutines menggunakan fungsi suspend untuk membuat urutan kode asinkron.

Coroutines adalah asynchronous

Coroutine berjalan secara independen dari langkah-langkah eksekusi utama program. Eksekusi ini bisa paralel atau pada prosesor terpisah. Bisa juga bahwa sementara sisa aplikasi sedang menunggu input, kita memasukkan sebuah bit pemrosesan. Salah satu aspek penting dari async adalah kita tidak dapat mengharapkan bahwa hasilnya tersedia, sampai kita secara eksplisit menunggu untuk itu.

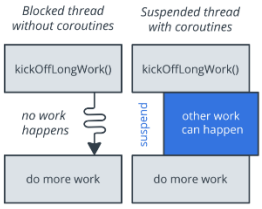
Misalnya, katakanlah kita memiliki pertanyaan yang memerlukan penelitian, dan kita meminta seorang kolega untuk menemukan jawabannya. Mereka pergi dan mengerjakannya, yang seperti mereka melakukan pekerjaan "secara tidak sinkron" dan "pada thread yang terpisah." kita dapat terus melakukan pekerjaan lain yang tidak bergantung pada jawabannya, sampai rekan kita kembali dan memberi tahu apa jawabannya.

Coroutines are non-blocking.

Non-blocking berarti coroutine tidak memblokir thread utama atau UI. Jadi dengan coroutine, pengguna selalu memiliki pengalaman semulus mungkin, karena interaksi UI selalu mendapat prioritas.

Coroutines menggunakan fungsi suspend untuk membuat urutan kode asinkron.

Penangguhan kata kunci adalah cara Kotlin menandai suatu fungsi, atau jenis fungsi, yang tersedia untuk coroutine. Ketika coroutine memanggil fungsi yang ditandai dengan menangguhkan, alih-alih memblokir sampai fungsi kembali seperti panggilan fungsi normal, coroutine menunda eksekusi hingga hasilnya siap. Kemudian coroutine dilanjutkan di tempat yang ditinggalkannya, dengan hasilnya. Sementara coroutine ditangguhkan dan menunggu hasilnya, ia membuka blokir thread-nya. Dengan begitu, fungsi atau coroutine lain dapat berjalan. Kata kunci yang ditangguhkan tidak menentukan thread bahwa kode berjalan. Fungsi suspend dapat berjalan pada thread latar belakang, atau pada thread utama.



Untuk menggunakan coroutine di Kotlin, Anda memerlukan tiga hal:

- sebuah job
- sebuah dispatcher
- scope

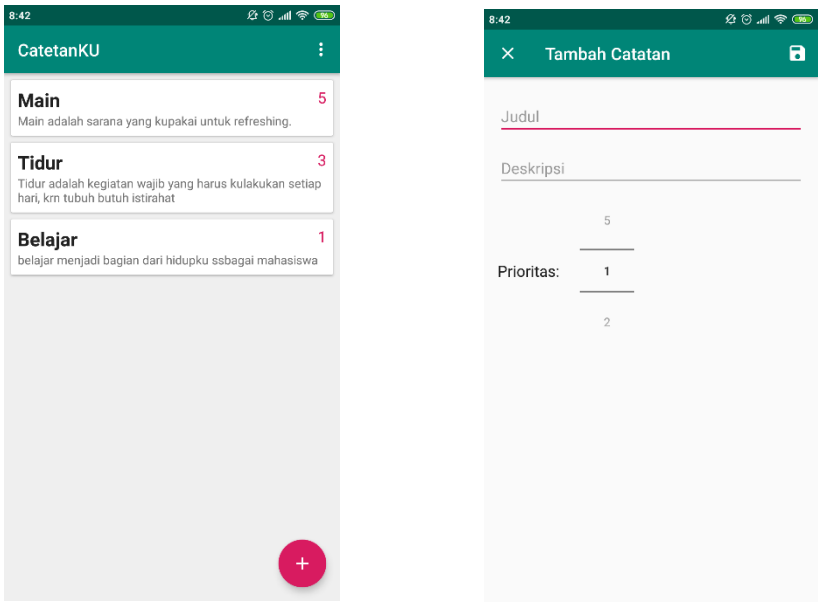
Job: Pada dasarnya, job adalah apa saja yang dapat dibatalkan. Setiap coroutine memiliki job, dan kita dapat menggunakan job itu untuk membatalkan coroutine. Job dapat diatur ke dalam hierarki orang parent-child. Membatalkan job dari parent segera membatalkan semua child job itu, yang jauh lebih nyaman daripada membatalkan setiap coroutine secara manual.

Dispatcher: Dispatcher mengirimkan coroutine untuk berjalan di berbagai thread. Misalnya, Dispatcher.Main menjalankan tugas di thread utama, dan Dispatcher.IO melepaskan muatan yang memblokir tugas I / O ke kumpulan thread yang dibagi.

Scope: Lingkup coroutine mendefinisikan konteks di mana coroutine berjalan. Lingkup menggabungkan informasi tentang pekerjaan dan operator koroutin. Cakupan melacak coroutine. Saat kita meluncurkan coroutine, itu "dalam scope," yang berarti bahwa kita telah menunjukkan scope yang akan melacak coroutine.

PRAKTIK

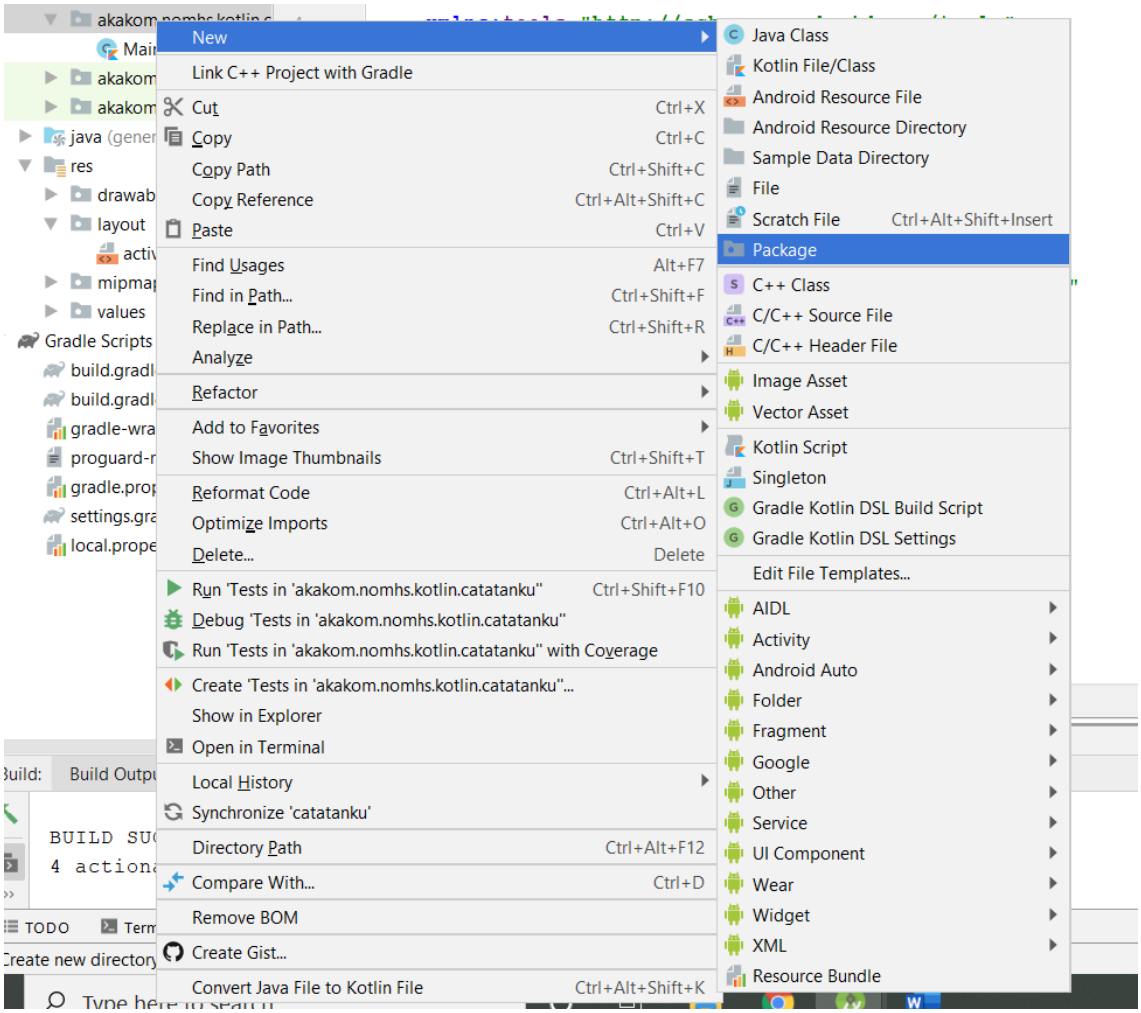
1. Kita akan membuat menu dengan desain sebagai berikut. Kita akan bekerja dengan satu table beberapa field.

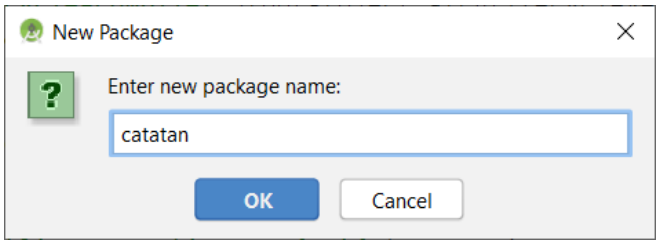


2. Layar pertama, ditunjukkan di sebelah kiri, digunakan untuk menampilkan data rekaman catatan yang sudah pernah dimasukkan pengguna. Layar kedua, ditunjukkan di sebelah kanan, digunakan untuk menambah catatan.
3. Buat project baru
4. Perbaharui Gradle seperti modul 9 (langkah 3)
5. Tambahkan komponen RecyclerView pada activity_main.xml

```
<androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/lightGray"
        tools:itemCount="5"
        tools:listitem="@layout/note_item" />
```

6. Buat package baru dibawa paket yang sudah ada, dengan nama **catatan**.





7. Buat kelas kotlin baru untuk kelas data dengan nama Note, dibawah package catatan.

```
@Entity(tableName = "note_table")
data class Note(
    var title: String,
    var description: String,
    var priority: Int
) {
    @PrimaryKey(autoGenerate = true)
    var id: Int = 0
}
```

8. Masih di dalam paket catatan, buat file DAO

```
@Dao
interface NoteDao {

    @Insert
    fun insert(note: Note)

    @Update
    fun update(note: Note)

    @Delete
    fun delete(note: Note)

    @Query("DELETE FROM note_table")
    fun deleteAllNotes()

    @Query("SELECT * FROM note_table ORDER BY priority DESC")
    fun getAllNotes(): LiveData<List<Note>>

}
```

9. Dalam paket catatan juga, buat file NoteDatabase

```
@Database(entities = [Note::class], version = 1)
abstract class NoteDatabase : RoomDatabase() {

    abstract fun noteDao(): NoteDao

    companion object {
        private var instance: NoteDatabase? = null

        fun getInstance(context: Context): NoteDatabase? {
            if (instance == null) {
                synchronized(NoteDatabase::class) {
                    instance = Room.databaseBuilder(
                        context.applicationContext,
                        NoteDatabase::class.java, "note_database"
                    )
                        .fallbackToDestructiveMigration()
                        .addCallback(roomCallback)
                        .build()
                }
            }
            return instance
        }

        fun destroyInstance() {
            instance = null
        }
```

```
        private val roomCallback = object : RoomDatabase.Callback() {
            override fun onCreate(db: SupportSQLiteDatabase) {
                super.onCreate(db)
                PopulateDbAsyncTask(instance)
                    .execute()
            }
        }
    }

    class PopulateDbAsyncTask(db: NoteDatabase?) : AsyncTask<Unit, Unit, Unit>() {
        private val noteDao = db?.noteDao()

        override fun doInBackground(vararg p0: Unit?) {
            noteDao?.insert(Note("Coba 1", "Deskripsi 1", 1))
        }
    }
}
```

10. Selanjutnya, masih dibawah package catatan, buatlah file kelas Kotlin bernama NoteRepository

```
class NoteRepository(application: Application) {

    private var noteDao: NoteDao

    private var allNotes: LiveData<List<Note>>

    init {
        val database: NoteDatabase = NoteDatabase.getInstance(
            application.applicationContext
        )!!
        noteDao = database.noteDao()
        allNotes = noteDao.getAllNotes()
    }

    fun insert(note: Note) {
        val insertNoteAsyncTask = InsertNoteAsyncTask(noteDao).execute(note)
    }

    fun update(note: Note) {
        val updateNoteAsyncTask = UpdateNoteAsyncTask(noteDao).execute(note)
    }

    fun delete(note: Note) {
        val deleteNoteAsyncTask = DeleteNoteAsyncTask(noteDao).execute(note)
    }

    fun deleteAllNotes() {
        val deleteAllNotesAsyncTask = DeleteAllNotesAsyncTask(
            noteDao
        ).execute()
    }

    fun getAllNotes(): LiveData<List<Note>> {
        return allNotes
    }

    companion object {
        private class InsertNoteAsyncTask(noteDao: NoteDao) : AsyncTask<Note, Unit, Unit>() {
            val noteDao = noteDao

            override fun doInBackground(vararg p0: Note?) {
                noteDao.insert(p0[0]!!)
            }
        }

        private class UpdateNoteAsyncTask(noteDao: NoteDao) : AsyncTask<Note, Unit, Unit>() {
            val noteDao = noteDao

            override fun doInBackground(vararg p0: Note?) {
                noteDao.update(p0[0]!!)
            }
        }

        private class DeleteNoteAsyncTask(noteDao: NoteDao) : AsyncTask<Note, Unit, Unit>() {
            val noteDao = noteDao

            override fun doInBackground(vararg p0: Note?) {
                noteDao.delete(p0[0]!!)
            }
        }

        private class DeleteAllNotesAsyncTask(noteDao: NoteDao) : AsyncTask<Unit, Unit, Unit>() {
            val noteDao = noteDao

            override fun doInBackground(vararg p0: Unit?) {
                noteDao.deleteAllNotes()
            }
        }
```

```
    }
}
```

11. Buat sebuah file class Kotlin dengan nama NoteViewMode di package utama.

```
class NoteViewModel(application: Application) : AndroidViewModel(application)
{
    private var repository: NoteRepository =
        NoteRepository(application)
    private var allNotes: LiveData<List<Note>> = repository.getAllNotes()

    fun insert(note: Note) {
        repository.insert(note)
    }

    fun update(note: Note) {
        repository.update(note)
    }

    fun delete(note: Note) {
        repository.delete(note)
    }

    fun deleteAllNotes() {
        repository.deleteAllNotes()
    }

    fun getAllNotes(): LiveData<List<Note>> {
        return allNotes
    }
}
```

12. Tambahkan layout pada layout/note_item.xml

```
<androidx.cardview.widget.CardView
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp">

    <RelativeLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="8dp">

        <TextView
                android:id="@+id/text_view_priority"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:text="1"
                android:textAppearance="@style/TextAppearance.AppCompat.Large"
                android:textColor="@color/colorAccent"
                android:textSize="18sp" />

        <TextView
                android:id="@+id/text_view_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentStart="true"
                android:layout_toStartOf="@id/text_view_priority"
                android:ellipsize="end"
                android:maxLines="1"
                android:text="Judul"
                android:textAppearance="@style/TextAppearance.AppCompat.Large"
                android:textStyle="bold" />
```

```
        <TextView
                android:id="@+id/text_view_description"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@id/text_view_title"
                android:text="Deskripsi"/>

    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

13. Buat vector Asset untuk symbol + (tambah), x (tutup) dan save (simpan)
14. Tambahkan sebuah FAB pada activity_main.xml

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/buttonAddNote"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_save_black_24dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:layout_gravity="bottom|right"/>
```

15. Buat adapter dengan nama NoteAdapter.

```
class NoteAdapter : ListAdapter<Note, NoteAdapter.NoteHolder>(DIFF_CALLBACK) {

    companion object {
        private val DIFF_CALLBACK = object : DiffUtil.ItemCallback<Note>() {
            override fun areItemsTheSame(oldItem: Note, newItem: Note): Boolean {
                return oldItem.id == newItem.id
            }

            override fun areContentsTheSame(oldItem: Note, newItem: Note): Boolean {
                return oldItem.title == newItem.title && oldItem.description == newItem.description
                        && oldItem.priority == newItem.priority
            }
        }
    }

    private var listener: OnItemClickListener? = null

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): NoteHolder {
        val itemView: View = LayoutInflater.from(parent.context).inflate(R.layout.note_item, parent, false)
        return NoteHolder(itemView)
    }

    override fun onBindViewHolder(holder: NoteHolder, position: Int) {
        val currentNote: Note = getItem(position)

        holder.textViewTitle.text = currentNote.title
        holder.textViewPriority.text = currentNote.priority.toString()
        holder.textViewDescription.text = currentNote.description
    }

    fun getNoteAt(position: Int): Note {
        return getItem(position)
    }

    inner class NoteHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        init {
            itemView.setOnClickListener {
                val position = adapterPosition
                if (position != RecyclerView.NO_POSITION) {
                    listener?.onItemClick(getItem(position))
                }
            }
        }

        var textViewTitle: TextView = itemView.text_view_title
        var textViewPriority: TextView = itemView.text_view_priority
        var textViewDescription: TextView = itemView.text_view_description
    }

    interface OnItemClickListener {
        fun onItemClick(note: Note)
    }

    fun setOnItemClickListener(listener: OnItemClickListener) {
        this.listener = listener
    }
}
```

16. Buat sebuah menu untuk menu utama pada menu/main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
            android:id="@+id/delete_all_notes"
            android:title="Hapus Semua Catatan"
            app:showAsAction="never"/>
</menu>
```

17. Buat sebuah menu untuk menu tambah pada menu/add_note_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
            android:id="@+id/save_note"
            android:icon="@drawable/ic_save"
            android:title="Simpan"
            app:showAsAction="ifRoom">
    </item>
</menu>
```

18. Buat sebuah activity untuk menambah data, dengan nama activity_add_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp">

    <EditText
            android:id="@+id/edit_text_title"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:hint="Judul"
            android:inputType="text"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

    <EditText
            android:id="@+id/edit_text_description"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:hint="Deskripsi"
            android:inputType="textMultiLine"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/edit_text_title" />

    <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Prioritas:"
            android:textAppearance="@android:style/TextAppearance.Medium"
            app:layout_constraintBottom_toBottomOf="@+id/number_picker_priority"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="@+id/number_picker_priority" />

    <NumberPicker
            android:id="@+id/number_picker_priority"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="24dp"
            android:layout_marginTop="8dp"
            app:layout_constraintStart_toEndOf="@+id/textView"
            app:layout_constraintTop_toBottomOf="@+id/edit_text_description" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

19. Buat kelas Kotlinnya AddEditNoteActivity.kt

```
class AddEditNoteActivity : AppCompatActivity() {
    companion object {
        const val EXTRA_ID = "com.piusanggoro.notesapp.EXTRA_ID"
        const val EXTRA_JUDUL = "com.piusanggoro.notesapp.EXTRA_JUDUL"
        const val EXTRA_DESKRIPSI = "com.piusanggoro.notesapp.EXTRA_DESKRIPSI"
        const val EXTRA_PRIORITAS = "com.piusanggoro.notesapp.EXTRA_PRIORITAS"
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_note)

        number_picker_priority.minValue = 1
        number_picker_priority.maxValue = 5

        supportActionBar?.setHomeAsUpIndicator(R.drawable.ic_close_black_24dp)

        if (intent.hasExtra(EXTRA_ID)) {
            title = "Edit Catatan"
            edit_text_title.setText(intent.getStringExtra(EXTRA_JUDUL))
            edit_text_description.setText(intent.getStringExtra(EXTRA_DESKRIPSI))
            number_picker_priority.value = intent.getIntExtra(EXTRA_PRIORITAS, 1)
        } else {
            title = "Tambah Catatan"
        }
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.add_note_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        return when (item?.itemId) {
            R.id.save_note -> {
                saveNote()
                true
            }
            else -> super.onOptionsItemSelected(item)
        }
    }

    private fun saveNote() {
        if (edit_text_title.text.toString().trim().isBlank() || edit_text_description.text.toString().trim().isBlank()) {
            Toast.makeText(this, "Catatan kosong!", Toast.LENGTH_SHORT).show()
            return
        }

        val data = Intent().apply {
            putExtra(EXTRA_JUDUL, edit_text_title.text.toString())
            putExtra(EXTRA_DESKRIPSI, edit_text_description.text.toString())
            putExtra(EXTRA_PRIORITAS, number_picker_priority.value)
            if (intent.getIntExtra(EXTRA_ID, -1) != -1) {
                putExtra(EXTRA_ID, intent.getIntExtra(EXTRA_ID, -1))
            }
        }

        setResult(Activity.RESULT_OK, data)
        finish()
    }
}
```

20. Daftarkan activity tersebut di AndroidManifest.xml

```
<activity android:name=".AddEditNoteActivity"
          android:parentActivityName=".MainActivity">
</activity>
```

21. PAda tag MainActivity, tambahkan juga launchMode pad, sehingga menjadi

```
<activity android:name=".MainActivity"
          android:launchMode="singleTop">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

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

22. Ubahlah activity_main.xml sehingga menjadi sebagai berikut. Tambahkan/ubah atribut yang belum ada (dikenali).

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/lightGray"
        tools:itemCount="5"
        tools:listitem="@layout/note_item" />


<com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/buttonAddNote"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_add_black_24dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:layout_gravity="bottom|right"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

23. Ubah MainActivity sehingga menjadi sebagai berikut

```
class MainActivity : AppCompatActivity() {

    companion object {
        const val ADD_NOTE_REQUEST = 1
        const val EDIT_NOTE_REQUEST = 2
    }

    private lateinit var noteViewModel: NoteViewModel

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        buttonAddNote.setOnClickListener {
            startActivityForResult(
                Intent(this, AddEditNoteActivity::class.java),
                ADD_NOTE_REQUEST
            )
        }

        recycler_view.layoutManager = LinearLayoutManager(this)
        recycler_view.setHasFixedSize(true)

        val adapter = NoteAdapter()
        recycler_view.adapter = adapter

        noteViewModel = ViewModelProviders.of(this).get(NoteViewModel::class.java)
        noteViewModel.getAllNotes().observe(this, Observer<List<Note>> {
            adapter.submitList(it)
        })

        ItemTouchHelper(object :
            ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT.or(ItemTouchHelper.RIGHT)) {
            override fun onMove(
                recyclerView: RecyclerView,
                viewHolder: RecyclerView.ViewHolder,
                target: RecyclerView.ViewHolder
            ): Boolean {
                return false
            }

            override fun onSwiped(viewHolder: RecyclerView.ViewHolder, direction: Int) {
                noteViewModel.delete(adapter.getNoteAt(viewHolder.adapterPosition))
                Toast.makeText(baseContext, "Catatan dihapus!", Toast.LENGTH_SHORT).show()
            }
        }
        ).attachToRecyclerView(recycler_view)

        adapter.setOnItemClickListener(object : NoteAdapter.OnItemClickListener {
            override fun onItemClick(note: Note) {
                val intent = Intent(baseContext, AddEditNoteActivity::class.java)
                intent.putExtra(AddEditNoteActivity.EXTRA_ID, note.id)
                intent.putExtra(AddEditNoteActivity.EXTRA_JUDUL, note.title)
                intent.putExtra(AddEditNoteActivity.EXTRA_DESKRIPSI, note.description)
```

```
                intent.putExtra(AddEditNoteActivity.EXTRA_PRIORITAS, note.priority)

                startActivityForResult(intent, EDIT_NOTE_REQUEST)
            }
        })
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.main_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        return when (item?.itemId) {
            R.id.delete_all_notes -> {
                noteViewModel.deleteAllNotes()
                Toast.makeText(this, "Semua sudah dihapus!", Toast.LENGTH_SHORT).show()
                true
            }
            else -> {
                super.onOptionsItemSelected(item)
            }
        }
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)

        if (requestCode == ADD_NOTE_REQUEST && resultCode == Activity.RESULT_OK) {
            val newNote = Note(
                data!!.getStringExtra(AddEditNoteActivity.EXTRA_JUDUL),
                data.getStringExtra(AddEditNoteActivity.EXTRA_DESKRIPSI),
                data.getIntExtra(AddEditNoteActivity.EXTRA_PRIORITAS, 1)
            )
            noteViewModel.insert(newNote)
            Toast.makeText(this, "Catatan disimpan!", Toast.LENGTH_SHORT).show()
        } else if (requestCode == EDIT_NOTE_REQUEST && resultCode == Activity.RESULT_OK) {
            val id = data?.getIntExtra(AddEditNoteActivity.EXTRA_ID, -1)

            if (id == -1) {
                Toast.makeText(this, "Pembaharuan gagal!", Toast.LENGTH_SHORT).show()
            }

            val updateNote = Note(
                data!!.getStringExtra(AddEditNoteActivity.EXTRA_JUDUL),
                data.getStringExtra(AddEditNoteActivity.EXTRA_DESKRIPSI),
                data.getIntExtra(AddEditNoteActivity.EXTRA_PRIORITAS, 1)
            )
            updateNote.id = data.getIntExtra(AddEditNoteActivity.EXTRA_ID, -1)
            noteViewModel.update(updateNote)
        } else {
            Toast.makeText(this, "Catatan tidak disimpan!", Toast.LENGTH_SHORT).show()
        }
    }
}
```

24. Jalankan dan amati hasilnya

LATIHAN

1. Modifikasilah aplikasi dengan menambahkan detil data pemilih nomor telepon.

TUGAS

1. Buat aplikasi baru dengan mengembangkan project diatas

REFERENSI

1. https://kotlinlang.org/docs/reference/
2. https://developer.android.com/kotlin
3. https://developer.android.com/courses/kotlin-android-fundamentals/toc
4. https://codelabs.developers.google.com/android-kotlin-fundamentals/
5. https://developer.android.com/kotlin/learn
6. https://developer.android.com/kotlin/resources