LAPORAN RESMI

MODUL III

INTENTS, MENU, DAN DIALOG

PEMROGRAMAN BERGERAK

NAMA	**: SEPTIYA YUTANTRI**
N.R.P	**: 200441100023**
DOSEn	**: ACHMAD DAFID, S.T., M.T**
ASISTEN	**: MUHAMMAD YAFIE ANWARY RAHMAN**
TGL PRAKTIKUM	**: 07 APRIL 2023**

Disetujui : 29 Mei 2023
Asisten

M. YAFIE ANWARY RAHMAN
190441100052

LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di era sekarang perkembangan teknologi sudah meningkat, manusia sudah tidak bisa dipisahkan dengan yang namanya teknologi. Untuk saat ini teknologi menjadi kebutuhan banyak orang. Dunia IT berkembang begitu pesat karena ditunjang dengan adanya perkembangan teknologi yang semakin canggih dan modern, pemanfaatan mobile learning yang dirasa perlu untuk menunjang proses belajar mengajar mahasiswa di pembelajaran Pemrograman Objek dan Perangkat Bergerak. Mobile Learning berbasis android dipilih mengingat hampir seluruh siswa di kelas memiliki handphone. Diharapkan dengan adanya mobile learning mampu meningkatkan semangat belajar karena didukung oleh teknologi yang bagus.

Pemebelajaran praktikum ini mempelajari tentang Intents, Menu, dan Dialog, dimana ketiga komponen tersebut merupakan dasar untuk membuat aplikasi yang mengintegrasikan beberapa komponen dan menjadi sebuah aplikasi yang fungsional. Intent merupakan suatu pesan yang digunakan untuk mengaktifkan tiga komponen dasar pada aplikasi Android yaitu Activity, Service, dan Broadcast Receiver. Aktifasi pada komponen-komponen tersebut bisa terjadi pada aplikasi yang sama atau berbeda, seperti menjalankan Activity, inisiasi Service, atau pengiriman pesan kepada Broadcast Receiver. Pada saat terjadi komunikasi antar komponen, Intent menyimpan paket informasi yang digunakan pada proses tersebut. Intent juga dapat digunakan untuk transfer data antar Activity. Pada saat sebuah Activity memanggil Activity yang lain, Intent dapat menyimpan data informasi yang ikut dikirimkan pada pemanggilan tersebut. Sedangkan untuk Dialog adalah jendela kecil yang meminta pengguna untuk membuat keputusan atau memasukkan informasi tambahan. Tampilan dialog tidak satu layar full dan biasanya digunakan untuk kejadian yang mengharuskan pengguna untuk melakukan aksi sebelum bisa melanjutkan proses selanjutnya. Penjelasan singkat tersebut harus dipahami dengan baik, sesuai arahan saat mengikuti praktikum.

Prasyarat untuk dapat mengikuti praktikum ini dengan baik adalah memiliki pengetahuan dalam Bahasa pemrograman berorientasi objek penuh seperti java,

C++, dan juga Kotlin. Untuk pengembangan, disarankan menggunakan sumber referensi selain modul praktikum ini, sehingga kitab bisa belajar lebih mendalam tentang materi yang terdapat di modul dari sumber lain.

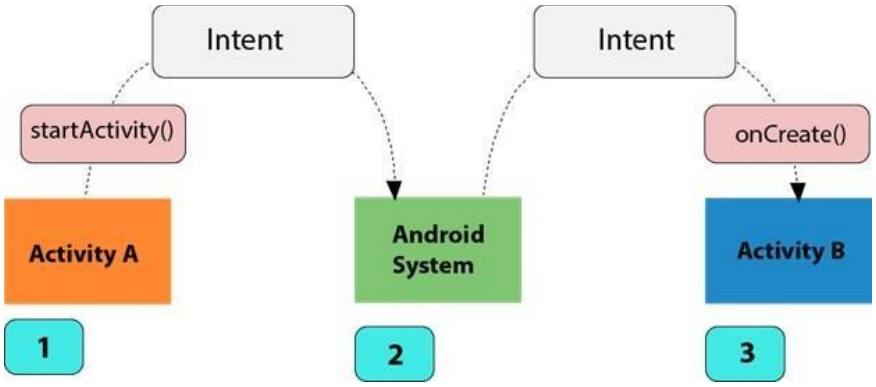
1.2 Tujuan

- Membuat Membuat sebuah Projek Aplikasi Android sederhana yang melibatkan komponen- komponen yang sudah dipelajari.
- Menggunakan menu dan Dialog.

BAB II
DASAR TEORI

2.1 Project Terintegrasi

Android menggunakan intents untuk melakukan pekerjaan tertentu di dalam aplikasinya. Begitu kita menguasai penggunaan intents, maka semua pengembangan aplikasi baru yang ada akan terbuka. Pada pertemuan ke 6 ini akan membahas tentang apa intents itu dan bagaimana dia digunakan. Intents adalah sebuah metode android untuk me-relay informasi tertentu dari satu aktivitas ke aktivitas lainnya. Secara lebih sederhana, Intents mengekspresikan kepada android untuk melakukan sesuatu.



Intent digunakan sebagai sebuah pesan yang dilewatkan diantara banyak aktivitas. Misalnya, mempunyai aktivitas yang mengharuskan untuk membuka web browser dan menampilkan sebuah halaman di perangkat android. Aktivitas kita akan mengirimkan "keinginan (intent) untuk membuka halaman x di web browser" yang dikenal dengan Intent WEB-SEARCH_ACTION, ke Android IntentResolver. IntentResolver mengurai melalui sebuah daftar Aktivitas dan memilih salah satu yang paling cocok dengan Intent, Lalu Intent Resolver mengirimkan halaman kita ke web browser dan memulai Web Browser Activity.

Intent dipecah ke dalam dua kategori utama:

- **Activity Action Intents** : Intent yang digunakan untuk memanggil Activity di luar aplikasi. Hanya satu Activity yang bisa ditangani oleh Intent. Misalnya saja untuk sebuah web browser, kita harus membuka Web Browser Activity untuk menampilkan halaman.
- **Broadcast Intents** : Intents yang dikirimkan untuk menangani lebih dari satu Activity. Contohnya, Broadcast intent yang akan menjadi sebuah pesan

yang dikirimkan oleh Android mengenai tingkat baterei saat itu. Banyak aktivitas bisa memproses Intent ini dan melakukan reaksi yang sesuai – misalnya, membatalkan sebuah Activity bila tingkat baterei berada di bawah titik tertentu.

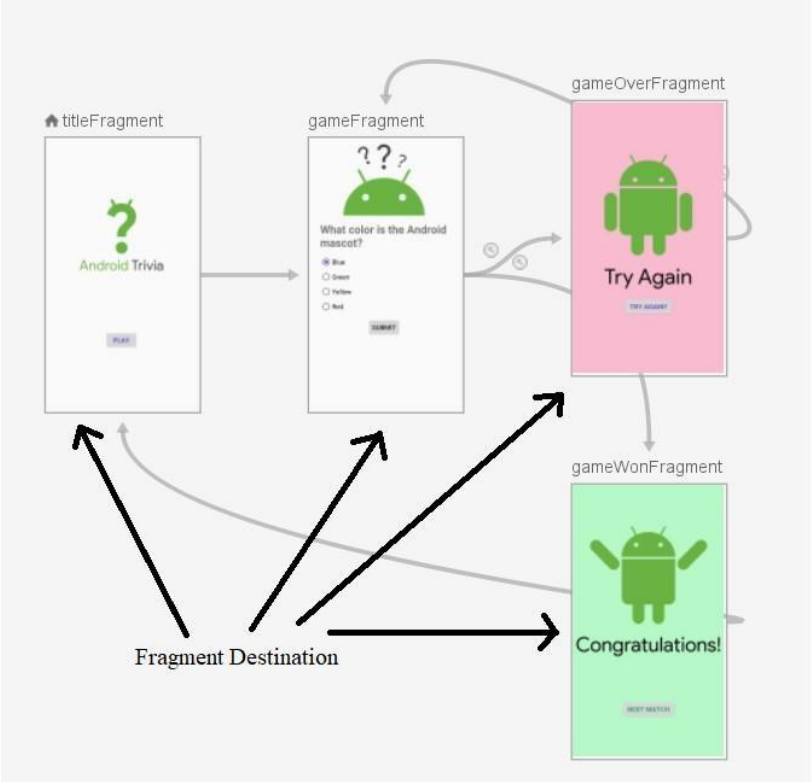
2.2 Menu dan Dialog

2.2.1 Komponen Navigasi

Navigation Graph (resource XML baru) adalah sumber daya yang berisi semua informasi terkait navigasi di satu lokasi terpusat, termasuk semua tempat di aplikasi, yang dikenal sebagai tujuan, dan kemungkinan jalur yang dapat dilalui pengguna melalui aplikasi. NavHostFragment (Layout XML view) adalah widget khusus yang ditambahkan ke layout, yang menampilkan berbagai tujuan dari Navigation Graph. NavController (objek Kotlin / Java) adalah objek yang melacak posisi saat ini dalam Navigation Graph, yang mengatur pertukaran konten tujuan di NavHostFragment saat kita bergerak melalui Navigation Graph. Saat kita menavigasi, kita akan menggunakan objek NavController, memberi tahu ke mana kita ingin pergi atau jalur apa yang ingin kita ambil dalam Navigation Graph. NavController kemudian akan menunjukkan tujuan yang sesuai di NavHostFragment.

2.2.2 Destinasi Navigation Graph

Komponen Navigasi memperkenalkan konsep destination. Destination adalah tempat apa pun yang dapat kita navigasi di aplikasi, biasanya sebuah fragment atau activity. Navigation Graph adalah jenis sumber daya baru yang mendefinisikan semua jalur yang mungkin seorang pengguna dapat ambil melalui aplikasi. Ini menunjukkan secara visual semua tujuan yang dapat dicapai dari tujuan tertentu. Android Studio menampilkan grafik di Editor Navigasinya. Contoh Navigation Graph adalah sebagai berikut.



Mengeksplorasi Navigation Editor

1) Buka res/navigation/mobile_navigation.xml
2) Klik Design untuk menuju mode Design.

Navigation graph menunjukkan destination yang ada.

Anatomi dari file navigation XML.

Semua perubahan yang dibuat di Editor Navigasi grafis mengubah file XML yang menyertainya, mirip dengan cara Editor Layout memodifikasi layout XML.

2.2.3 Activities dan Navigation

Komponen Navigasi mengikuti panduan yang dijabarkan dalam Prinsip Navigasi. Prinsip Navigasi merekomendasikan kita menggunakan aktivitas sebagai

titik masuk untuk aplikasi kita. Aktifitas juga akan berisi navigasi global, seperti bottom nav. Sebagai perbandingan, fragment akan menjadi layout spesifik tujuan yang sebenarnya. Agar semua ini berfungsi, kita perlu memodifikasi layout aktivitas kita untuk memuat widget khusus yang disebut NavHostFragment. NavHostFragment menukar destination fragment yang berbeda masuk dan keluar saat kita menavigasi navigation graph.

BAB III
TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan apa yang kamu ketahui tentang Intent
2. Sebutkan dan jelaskan 2 kategori utama tentang Intent.

3.2 Jawaban.

1. Intent adalah sebuah kelas dalam pemrograman Android yang berfungsi untuk perpindahan halaman. Intent berfungsi sebagai sebuah jembatan yang menghubungkan interaksi antar Activity di aplikasi Android.
   Intinya Intent merupakan mekanisme untuk melakukan sebuah action dan komunikasi antar komponen aplikasi

2. Intent dipecah ke dalam 2 kategori utama yaitu.
   - Activity Action Intents : Intent yang digunakan untuk memanggil Activity diluar aplikasi. Hanya satu activity yang ditangani oleh Intent. Misalnya untuk sebuah web browser, harus membuka web Browser Activity untuk menampilkan halaman.
   - Broadcast Intents : Intents yang dikirimkan untuk menangani lebih dari satu Activity. Contohnya Broadcast Intent yang akan menjadi sebuah peran yang akan dikirimkan oleh Android mengenai tingkat baterai saat itu.

BAB IV

IMPLEMENTASI

4.1 SourCode

a. AndroidManifest

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

    <application
        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=".activity.EditActivity"></activity>
        <activity android:name=".activity.MainActivity">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

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

</manifest>
```

b. EditActivity

```
package com.lazday.kotlinroommvvm.activity

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import com.lazday.kotlinroommvvm.R
import com.lazday.kotlinroommvvm.room.Constant
import com.lazday.kotlinroommvvm.room.Note
import com.lazday.kotlinroommvvm.room.NoteDB
import kotlinx.android.synthetic.main.activity_edit.*
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch

class EditActivity : AppCompatActivity() {

    private val db by lazy { NoteDB(this) }
    private var noteId = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.activity_edit)
        setupView()
        setupLstener()
    }

    private fun setupView(){

supportActionBar!!.setDisplayHomeAsUpEnabled(true)
        when (intentType()) {
            Constant.TYPE_CREATE -> {
                supportActionBar!!.title = "BUAT BARU"
                button_save.visibility = View.VISIBLE
                button_update.visibility = View.GONE
            }
            Constant.TYPE_READ -> {
                supportActionBar!!.title = "BACA"
                button_save.visibility = View.GONE
                button_update.visibility = View.GONE
                getNote()
            }
            Constant.TYPE_UPDATE -> {
                supportActionBar!!.title = "EDIT"
                button_save.visibility = View.GONE
                button_update.visibility = View.VISIBLE
                getNote()
            }
        }
    }

    private fun setupLstener(){
        button_save.setOnClickListener {
            CoroutineScope(Dispatchers.IO).launch {
                db.noteDao().addNote(
                    Note(
                        0,
                        edit_title.text.toString(),
                        edit_note.text.toString()
                    )
                )
                finish()
            }
        }
        button_update.setOnClickListener {
            CoroutineScope(Dispatchers.IO).launch {
                db.noteDao().updateNote(
                    Note(
                        noteId,
                        edit_title.text.toString(),
                        edit_note.text.toString()
                    )
                )
                finish()
            }
        }
    }

    private fun getNote(){
        noteId = intent.getIntExtra("note_id", 0)
        CoroutineScope(Dispatchers.IO).launch {
```

```
            val notes =
db.noteDao().getNote(noteId).get(0)
            edit_title.setText( notes.title )
            edit_note.setText( notes.note )
        }
    }

    override fun onSupportNavigateUp(): Boolean {
        finish()
        return super.onSupportNavigateUp()
    }

    private fun intentType(): Int {
        return intent.getIntExtra("intent_type", 0)
    }
}
```

c. MainActivity

```
package com.lazday.kotlinroommvvm.activity

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.appcompat.app.AlertDialog
import androidx.recyclerview.widget.LinearLayoutManager
import com.lazday.kotlinroommvvm.R
import com.lazday.kotlinroommvvm.room.Constant
import com.lazday.kotlinroommvvm.room.Note
import com.lazday.kotlinroommvvm.room.NoteDB
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.coroutines.*

class MainActivity : AppCompatActivity() {

    private val db by lazy { NoteDB(this) }
    lateinit var noteAdapter: NoteAdapter

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

    override fun onResume() {
        super.onResume()
        loadData()
    }

    private fun loadData(){
        CoroutineScope(Dispatchers.IO).launch {
            noteAdapter.setData(db.noteDao().getNotes())
            withContext(Dispatchers.Main) {
                noteAdapter.notifyDataSetChanged()
            }
        }
```

```
    }

    private fun setupView (){
        supportActionBar!!.apply {
            title = "Catatan"
        }
    }

    private fun setupListener(){
        button_create.setOnClickListener {
            intentEdit(Constant.TYPE_CREATE, 0)
        }
    }

    private fun setupRecyclerView () {

        noteAdapter = NoteAdapter(
            arrayListOf(),
            object : NoteAdapter.OnAdapterListener {
                override fun onClick(note: Note) {
                    intentEdit(Constant.TYPE_READ,
note.id)
                }

                override fun onUpdate(note: Note) {
                    intentEdit(Constant.TYPE_UPDATE,
note.id)
                }

                override fun onDelete(note: Note) {
                    deleteAlert(note)
                }

            })

        list_note.apply {
            layoutManager =
LinearLayoutManager(applicationContext)
            adapter = noteAdapter
        }

    }

    private fun intentEdit(intent_type: Int, note_id:
Int) {
        startActivity(
            Intent(this, EditActivity::class.java)
                .putExtra("intent_type", intent_type)
                .putExtra("note_id", note_id)
        )

    }

    private fun deleteAlert(note: Note){
        val dialog = AlertDialog.Builder(this)
        dialog.apply {
            setTitle("Konfirmasi Hapus")
            setMessage("Yakin hapus ${note.title}?")
            setNegativeButton("Batal") { dialogInterface,
```

```
i ->
                dialogInterface.dismiss()
            }
            setPositiveButton("Hapus") { dialogInterface,
i ->
                CoroutineScope(Dispatchers.IO).launch {
                    db.noteDao().deleteNote(note)
                    dialogInterface.dismiss()
                    loadData()
                }
            }
        }

        dialog.show()
    }
}
```

d. NoteAdapter

```
package com.lazday.kotlinroommvvm.activity

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.lazday.kotlinroommvvm.R
import com.lazday.kotlinroommvvm.room.Note
import kotlinx.android.synthetic.main.adapter_main.view.*

class NoteAdapter (var notes: ArrayList<Note>, var
listener: OnAdapterListener) :
    RecyclerView.Adapter<NoteAdapter.NoteViewHolder>(){

    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): NoteViewHolder {
        return NoteViewHolder(
            LayoutInflater.from(parent.context)
                .inflate(
                    R.layout.adapter_main,
                    parent,
                    false
                )
        )
    }

    override fun getItemCount() = notes.size

    override fun onBindViewHolder(holder: NoteViewHolder,
position: Int) {
        val note = notes[position]
        holder.view.text_title.text = note.title
        holder.view.text_title.setOnClickListener {
            listener.onClick(note)
        }
        holder.view.icon_edit.setOnClickListener {
            listener.onUpdate(note)
        }
        holder.view.icon_delete.setOnClickListener {
```

```
                listener.onDelete(note)
            }
        }

        class NoteViewHolder(val view: View) :
    RecyclerView.ViewHolder(view)

        fun setData(newList: List<Note>) {
            notes.clear()
            notes.addAll(newList)
        }

        interface OnAdapterListener {
            fun onClick(note: Note)
            fun onUpdate(note: Note)
            fun onDelete(note: Note)
        }
    }
```

e. Constant

```
package com.lazday.kotlinroommvvm.room

class Constant {
    companion object {
        const val TYPE_READ     = 0
        const val TYPE_CREATE   = 1
        const val TYPE_UPDATE   = 2
    }
}
```

f. Note

```
package com.lazday.kotlinroommvvm.room

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity
data class Note(
    @PrimaryKey(autoGenerate = true)
    val id : Int = 0,
    val title: String,
    val note: String
)
```

g. NoteDao

```
package com.lazday.kotlinroommvvm.room

import androidx.room.*

@Dao
interface NoteDao {
    @Insert
    suspend fun addNote(note: Note)
```

```
    @Query("SELECT * FROM note ORDER BY id DESC")
    suspend fun getNotes() : List<Note>

    @Query("SELECT * FROM note WHERE id=:note_id")
    suspend fun getNote(note_id: Int) : List<Note>

    @Update
    suspend fun updateNote(note: Note)

    @Delete
    suspend fun deleteNote(note: Note)
}
```

h. NoteDB

```
package com.lazday.kotlinroommvvm.room

import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

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

    abstract fun noteDao() : NoteDao

    companion object {

        @Volatile private var instance : NoteDB? = null
        private val LOCK = Any()

        operator fun invoke(context: Context) = instance
?: synchronized(LOCK){
            instance ?: buildDatabase(context).also {
                instance = it
            }
        }

        private fun buildDatabase(context: Context) =
Room.databaseBuilder(
            context.applicationContext,
            NoteDB::class.java,
            "note12345.db"
        ).build()

    }
}
```

i. ExampleInstrumentTest

```
package com.lazday.kotlinroommvvm

import androidx.test.platform.app.InstrumentationRegistry
import androidx.test.ext.junit.runners.AndroidJUnit4
```

```
import org.junit.Test
import org.junit.runner.RunWith

import org.junit.Assert.*

/**
 * Instrumented test, which will execute on an Android
device.
 *
 * See [testing
documentation](http://d.android.com/tools/testing).
 */
@RunWith(AndroidJUnit4::class)
class ExampleInstrumentedTest {
    @Test
    fun useAppContext() {
        // Context of the app under test.
        val appContext =
InstrumentationRegistry.getInstrumentation().targetContex
t
        assertEquals("com.lazday.kotlinroommvvm",
appContext.packageName)
    }
}
```

j. ExampleUnirTest

```
package com.lazday.kotlinroommvvm

import org.junit.Test

import org.junit.Assert.*

/**
 * Example local unit test, which will execute on the
development machine (host).
 *
 * See [testing
documentation](http://d.android.com/tools/testing).
 */
class ExampleUnitTest {
    @Test
    fun addition_isCorrect() {
        assertEquals(4, 2 + 2)
    }
}
```

k. Activity_edit

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    tools:context=".activity.EditActivity"
    android:padding="20dp"
    >

    <EditText
        android:id="@+id/edit_title"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Judul"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        />
    <EditText
        android:id="@+id/edit_note"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Tulis Catatan"
        android:minLines="3"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintTop_toBottomOf="@+id/edit_title"
        android:layout_marginTop="10dp"
        android:gravity="top"
        />
    <Button
        android:id="@+id/button_save"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="SAVE"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintTop_toBottomOf="@+id/edit_note"
        android:layout_marginTop="20dp"
        />
    <Button
        android:id="@+id/button_update"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="UPDATE"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintTop_toBottomOf="@+id/button_save"
        android:layout_marginTop="20dp"
        />

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

l. Activity_main

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

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/list_note"
        android:layout_width="0dp"
        android:layout_height="0dp"

app:layout_constraintBottom_toTopOf="@+id/button_create"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:listitem="@layout/adapter_main"
        />

    <Button
        android:id="@+id/button_create"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Tulis Catatan"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_margin="10dp"
        />

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

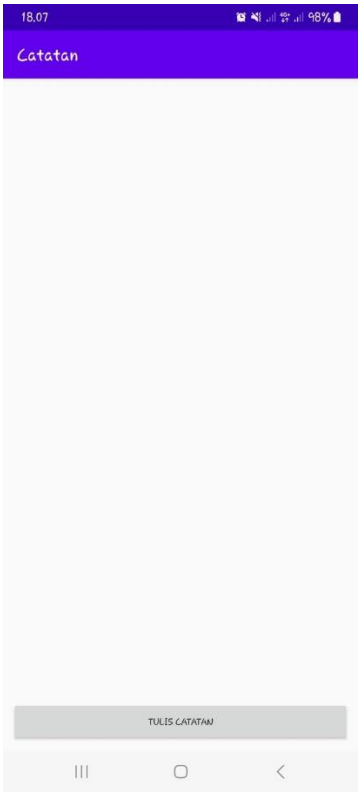
m. Adapter_main

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:padding="10dp">

    <TextView
        android:id="@+id/text_title"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        tools:text="Nanti kita cerita hari ini"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toStartOf="@+id/icon_edit"
        />
    <ImageView
        android:id="@+id/icon_edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_edit"
```

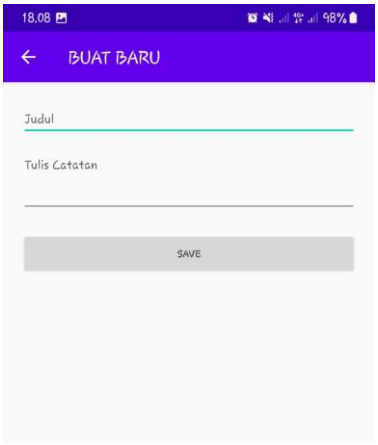
```
            android:padding="10dp"
            app:layout_constraintTop_toTopOf="parent"

    app:layout_constraintEnd_toStartOf="@+id/icon_delete"
            />
        <ImageView
            android:id="@+id/icon_delete"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_delete"
            android:padding="10dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            />

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

4.2 Hasil Run

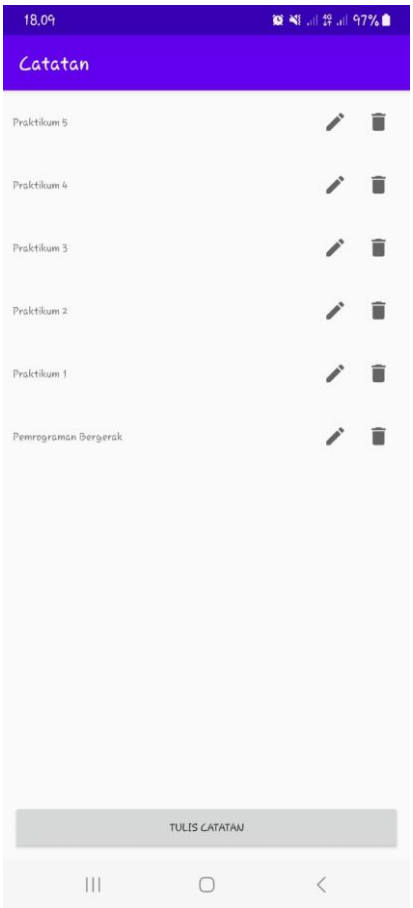
a. Pada gambar pertama hasil run berikut menunjukkan tampilan awal atau fragment 1 pada sistem yang telah dibuat



b. Gambar kedua hasil run menunjukkan jika setelah kita mengklik tulis catatan, maka akan muncul untuk menulis catatan yang kita inginkan.

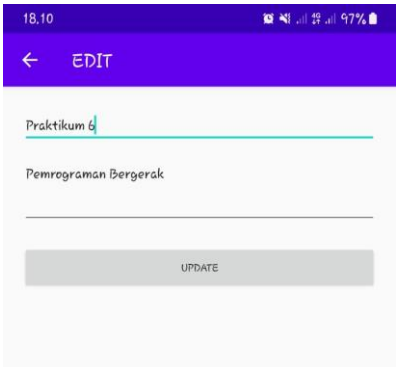


c. Pada gambar hasil run ketiga berikut, menunjukkan bahwa setelah kita memasukkan beberapa catatan akan muncul seperti gambar tersebut yang juga terdapat gambar pensil dan sampah, dimana pensil sendiri digunakan untuk mengubah catatan tersebut sesuai yang kita inginkan.

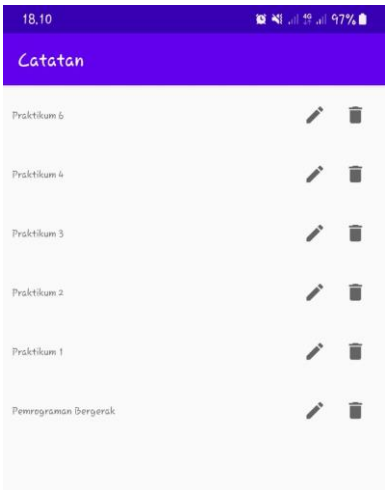


d. Gambar hasil run ke empat ini menunjukkan bahwa jika setelah mengklik tanda pensil seperti yang ada pada keterangan gambar ketiga,

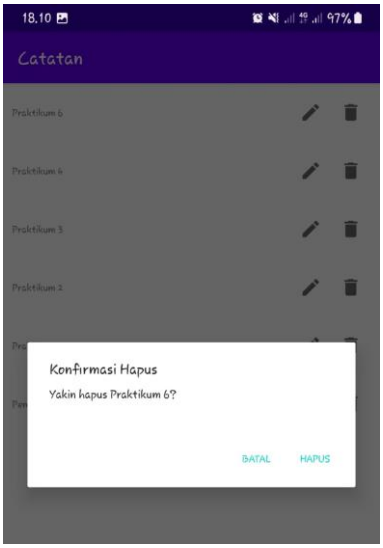
maka akan muncul halaman berikut, dimana kita dapat mengubah catatan sesuai yang kita inginkan, jika sudah selesai kemudian klik update dibagian bawahnya.



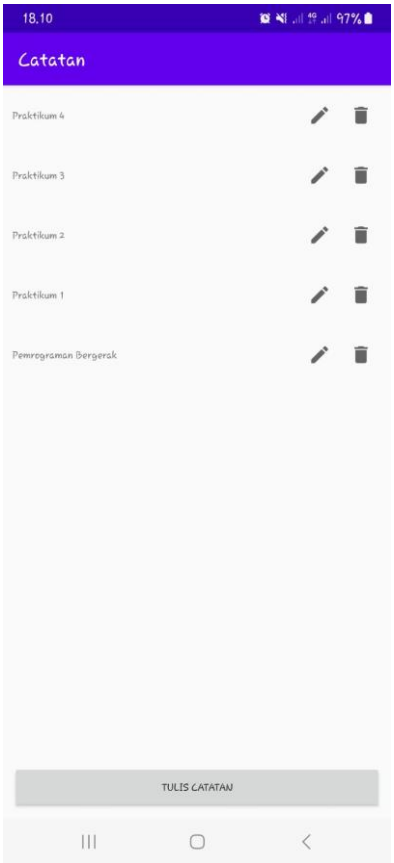
e. Pada gambar ke lima menunjukkan bahwa setelah catatan diupdate atau diedit maka akan tampil seperti semula, tapi yang awalnya bertuliskan Praktikum 5 menjadi Praktikum 6 seperti gambar dibawah ini.



f. Pada gambar ke enam dibawah ini menunjukkan bahwa setelah kita mengklik gambar sampah yang ada disebelah kanan gambar pensil, maka akan menampilkan sebuah peringatan yang bertuliskan “konfirmasi hapus” dimana terdapat pilihan “hapus” dan “batal” jika kita ingin menghapus catatan tersebut, atau batal menghapusnya.



g. Terakhir gambar hasil run dibawah ini menunjukkan bahwa pada tulisan “Praktikum 5” yang di update menjadi “Praktikum 6” di hapus, catatan yang tersedia atau terdapat pada gambar hanya sampai pada “Praktikum 4”



BAB V
PENUTUP

5.1 Analisa

Modul yang mempelajari tentang intens, menu, dan dialog di Android Studio adalah modul yang sangat penting untuk dipelajari bagi para pengembang aplikasi Android. Dalam modul ini, pengguna akan belajar cara membuat dan mengelola intens, menu, dan dialog di dalam aplikasi Android menggunakan Android Studio. Intens adalah cara untuk menghubungkan antara komponen aplikasi Android. Dalam modul ini, pengguna akan belajar cara membuat dan mengelola intens untuk memfasilitasi komunikasi antara berbagai bagian dari aplikasi. Menu adalah elemen penting dari antarmuka pengguna di aplikasi Android. Dalam modul ini, pengguna akan belajar cara membuat dan mengelola menu, serta cara menambahkan tindakan pada menu tersebut. Dialog adalah jendela kecil yang muncul di atas antarmuka pengguna utama. Dalam modul ini, pengguna akan belajar cara membuat dan mengelola dialog, serta cara mengkustomisasi tampilan dialog sesuai dengan kebutuhan aplikasi. Secara keseluruhan, modul ini akan memberikan pengguna pemahaman yang lebih mendalam tentang cara memanfaatkan intens, menu, dan dialog untuk meningkatkan kualitas dan fungsionalitas aplikasi Android yang dibuat.

5.2 Kesimpulan

Dari modul 3 ini kami dapat mempelajari tentang intens, menu, dan dialog di Android Studio, dan dapat disimpulkan bahwasannya ketiga komponen tersebut sangat penting untuk membangun aplikasi Android yang berkualitas. Intens memungkinkan komunikasi antara berbagai bagian dari aplikasi, menu memberikan akses mudah ke fitur-fitur penting, dan dialog memfasilitasi interaksi dengan pengguna secara lebih terfokus. Dalam modul ini, pengguna akan belajar cara membuat dan mengelola intens, menu, dan dialog menggunakan Android Studio. Mereka juga akan mempelajari cara mengkustomisasi tampilan dan menambahkan tindakan pada menu serta cara mengatur tampilan dan perilaku dari dialog. Dengan memahami intens, menu, dan dialog, pengguna dapat meningkatkan fungsionalitas dan kualitas aplikasi Android yang dibuat serta memberikan pengalaman pengguna

yang lebih baik. Oleh karena itu, modul ini sangat penting untuk dipelajari oleh para pengembang aplikasi Android.