

LAPORAN RESMI
MODUL V
FRAGMENT
PEMROGRAMAN BERGERAK



NAMA	: SEPTIYA YUTANTRI
N.R.P	: 200441100023
DOSEn	: ACHMAD DAFID, S.T., M.T
ASISTEN	: MUHAMMAD YAFIE ANWARY RAHMAN
TGL PRAKTIKUM	: 11 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.

Pembelajaran praktikum ini mempelajari tentang Fragment, dimana Android memperkenalkan fragmen di Android 3.0 (API level 11), terutama untuk mendukung desain UI yang lebih dinamis dan fleksibel pada layar besar, seperti tablet. Karena layar tablet jauh lebih besar daripada layar handset, maka lebih banyak ruang untuk mengombinasikan dan bertukar komponen UI. Fragmen memungkinkan desain seperti itu tanpa perlu mengelola perubahan kompleks pada hierarki tampilan. Dengan membagi layout aktivitas menjadi beberapa fragmen, kita bisa mengubah penampilan aktivitas saat waktu proses dan mempertahankan perubahan itu di back-stack yang dikelola oleh aktivitas. Mode-mode tersebut kini tersedia secara luas melalui library dukungan fragmen. Kita harus mendesain masing-masing fragmen sebagai komponen aktivitas modular dan bisa digunakan kembali. Yakni, karena setiap fragmen mendefinisikan layoutnya dan perilakunya dengan callback daur hidupnya sendiri, kita bisa memasukkan satu fragmen dalam banyak aktivitas, sehingga kita harus mendesainnya untuk digunakan kembali dan mencegah memanipulasi satu fragmen dari fragmen lain secara langsung. Ini terutama penting karena dengan fragmen modular kita bisa mengubah kombinasi fragmen untuk ukuran layar yang berbeda. Saat mendesain aplikasi untuk mendukung tablet maupun handset, kita bisa menggunakan kembali fragmen dalam konfigurasi layout yang berbeda untuk mengoptimalkan pengalaman pengguna

berdasarkan ruang layar yang tersedia. Misalnya, pada handset, fragmen mungkin perlu dipisahkan untuk menyediakan UI panel tunggal bila lebih dari satu yang tidak cocok dalam aktivitas yang sama. 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 Proyek Aplikasi Android sederhana yang melibatkan komponen- komponen yang sudah dipelajari.
- Menggunakan menu dan Dialog.

BAB II

DASAR TEORI

2.1 Fragment

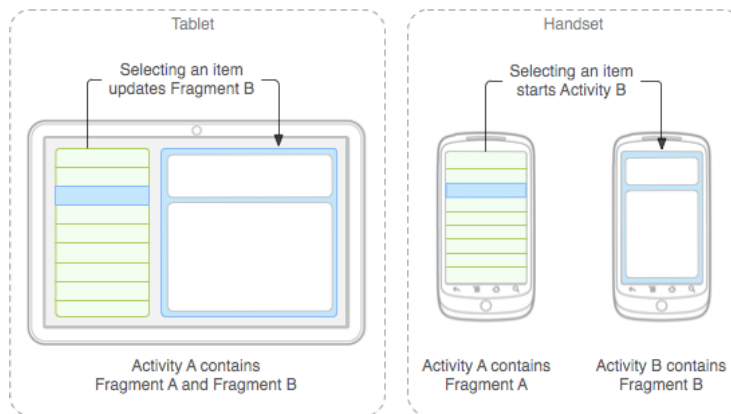
Fragment adalah bagian UI aplikasi Anda yang dapat digunakan kembali. Sebuah fragmen menentukan dan mengelola tata letaknya sendiri, memiliki siklus proses sendiri, serta dapat menangani peristiwa inputnya sendiri. Fragmen tidak dapat berjalan sendiri. Fragmen harus dihosting oleh aktivitas atau fragmen lain. Hierarki tampilan fragmen menjadi bagian dari, atau dilampirkan ke, hierarki tampilan host.

2.2 Filosofi

Android memperkenalkan fragmen di Android 3.0 (API level 11), terutama untuk mendukung desain UI yang lebih dinamis dan fleksibel pada layar besar, seperti tablet. Karena layar tablet jauh lebih besar daripada layar handset, maka lebih banyak ruang untuk mengombinasikan dan bertukar komponen UI. Fragmen memungkinkan desain seperti itu tanpa perlu mengelola perubahan kompleks pada hierarki tampilan. Dengan membagi layout aktivitas menjadi beberapa fragmen, kita bisa mengubah penampilan aktivitas waktu proses dan mempertahankan perubahan itu di back-stack yang dikelola oleh aktivitas. Mode-mode tersebut kini tersedia secara luas melalui library dukungan fragmen. Misalnya, aplikasi berita bisa menggunakan satu fragmen untuk menampilkan daftar artikel disebelah kiri dan fragmen lainnya untuk menampilkan artikel di sebelah kanan—kedua fragmen ini muncul di satu aktivitas, berdampingan, dan masing-masing fragmen memiliki serangkaian metode callback daur hidup dan menangani kejadian masukan pengguna sendiri. Sehingga, sebagai ganti menggunakan satu aktivitas untuk memilih artikel dan aktivitas lainnya untuk membaca artikel, pengguna bisa memilih artikel dan membaca semuanya dalam aktivitas yang sama, sebagaimana diilustrasikan dalam layout tablet pada gambar 1.

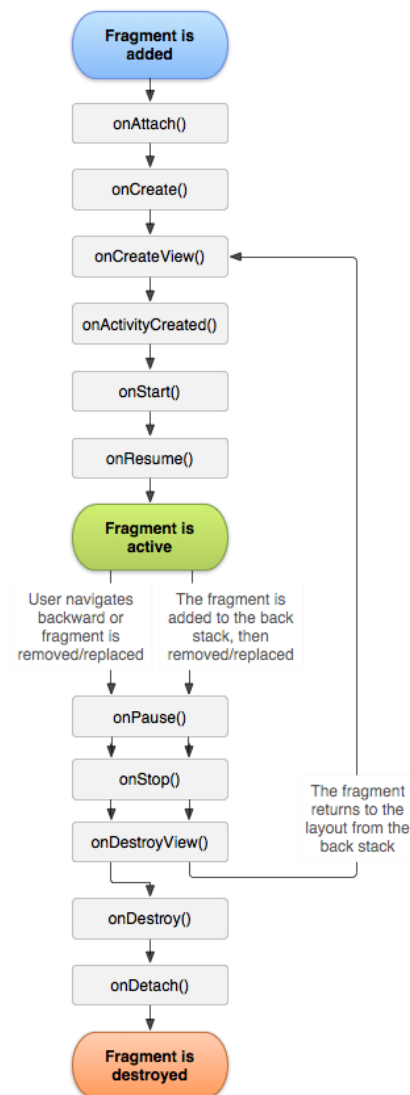
Kita harus mendesain masing-masing fragmen sebagai komponen aktivitas modular dan bisa digunakan kembali. Yakni, karena setiap fragmen mendefinisikan layoutnya dan perilakunya dengan callback daur hidupnya sendiri, kita bisa memasukkan satu fragmen dalam banyak aktivitas, sehingga kita harus mendesainnya untuk digunakan kembali dan mencegah memanipulasi satu fragmen dari fragmen lain secara langsung. Ini terutama

penting karena dengan fragmen modular kita bisa mengubah kombinasi fragmen untuk ukuran layar yang berbeda. Saat mendesain aplikasi untuk mendukung tablet maupun handset, kita bisa menggunakan kembali fragmen dalam konfigurasi layout yang berbeda untuk mengoptimalkan pengalaman pengguna berdasarkan ruang layar yang tersedia. Misalnya, pada handset, fragmen mungkin perlu dipisahkan untuk menyediakan UI panel tunggal bila lebih dari satu yang tidak cocok dalam aktivitas yang sama.



Gambar 1. Contoh bagaimana dua modul UI yang ditentukan oleh fragmen bisa digabungkan ke dalam satu aktivitas untuk desain tablet, namun dipisahkan untuk desain handset.

Misalnya—untuk melanjutkan contoh aplikasi berita —aplikasi bisa menyematkan dua fragmen dalam *Aktivitas A*, saat berjalan pada perangkat berukuran tablet. Akan tetapi, pada layar berukuran handset, ruang untuk kedua fragmen tidak sehingga *Aktivitas A* hanya menyertakan fragmen untuk daftar artikel, dan saat pengguna memilih artikel, *Aktivitas B* akan dimulai, termasuk fragmen kedua untuk membaca artikel. Sehingga, aplikasi mendukung tablet dan handset dengan menggunakan kembali fragmen dalam kombinasi berbeda, seperti diilustrasikan dalam gambar 1.



Gambar 2. Daur hidup fragmen (saat aktivitasnya berjalan).

2.3 Membuat Fragmen

Untuk membuat fragmen, kita harus membuat subclass `Fragment` (atau subclass-nya yang ada). Class `Fragment` memiliki kode yang mirip seperti `Activity`. Class ini memiliki metode callback yang serupa dengan aktivitas, seperti `onCreate()`, `onStart()`, `onPause()`, dan `onStop()`. Sebenarnya, jika kita mengonversi aplikasi Android saat ini untuk menggunakan fragmen, kita mungkin cukup memindahkan kode dari metode callback aktivitas ke masing-masing metode callback fragmen. Biasanya, kita harus mengimplementasikan setidaknya metode daur hidup berikut ini:

[onCreate\(\)](#) : Sistem akan memanggilnya saat membuat fragmen. Dalam implementasi, kita harus melakukan inisialisasi komponen penting dari fragmen yang ingin dipertahankan saat fragmen dihentikan sementara atau dihentikan, kemudian dilanjutkan.

[onCreateView\(\)](#) : Sistem akan memanggilnya saat fragmen menggambar antarmuka pengguna untuk yang pertama kali. Untuk menggambar UI fragmen, kita harus mengembalikan View dari metode ini yang menjadi root layout fragmen. Hasil yang dikembalikan bisa berupa null jika fragmen tidak menyediakan UI.

[onPause\(\)](#) : Sistem akan memanggil metode ini sebagai indikasi pertama bahwa pengguna sedang meninggalkan fragmen kita (walau itu tidak selalu berarti fragmen sedang dimusnahkan). Di sinilah biasanya kita harus mengikat perubahan yang harus dipertahankan di luar sesi pengguna saat ini (karena pengguna mungkin tidak akan kembali).

Kebanyakan aplikasi harus mengimplementasikan setidaknya tiga metode ini untuk setiap fragmen, tetapi ada beberapa metode callback lain yang juga harus kita gunakan untuk menangani berbagai tahap daur hidup fragmen. Perhatikan bahwa kode yang mengimplementasikan aksi daur hidup dari komponen dependen harus ditempatkan di komponen itu sendiri, bukan dalam implementasi callback fragmen

2.4 Menambahkan antarmuka pengguna

Fragmen biasanya digunakan sebagai bagian dari antarmuka pengguna aktivitas dan menyumbangkan layoutnya sendiri ke aktivitas. Untuk menyediakan layout fragmen, kita harus mengimplementasikan metode callback `onCreateView()`, yang dipanggil sistem Android bila tiba saatnya fragmen menggambar layoutnya. Implementasi kita atas metode ini harus mengembalikan View yang menjadi root layout fragmen. Untuk mengembalikan layout dari `onCreateView()`, Kita bisa memekarkannya dari resource layout yang

ditentukan di XML. Untuk membantu melakukannya, `onCreateView()` menyediakan objek `LayoutInflater`.

Misalnya, terdapat subclass `Fragment` yang memuat layout dari file

`example_fragment.xml`: `class ExampleFragment : Fragment() {`

`override fun onCreateView(inflater:LayoutInflater, container:`

`ViewGroup?, savedInstanceState: Bundle?`

```

): View {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.example_fragment, container, false)
}
}

```

Parameter container yang diteruskan ke onCreateView() adalah induk ViewGroup (dari layout aktivitas) tempat layout fragmen akan disisipkan. Parameter savedInstanceState adalah Bundle yang menyediakan data tentang instance fragmen sebelumnya, jika fragmendilanjutkan.

Metode inflate() mengambil tiga argumen:

1. ID sumber daya layout yang ingin dimekarkan.
2. ViewGroup akan menjadi induk dari layout yang dimekarkan. container perlu diteruskan agar sistem menerapkan parameter layout ke tampilan akar layout yang dimekarkan, yang ditetapkan dalam tampilan induk yang akan dituju.
3. Boolean yang menunjukkan apakah layout yang dimekarkan harus dilampirkan ke ViewGroup (parameter kedua) selama pemekaran. (Dalam hal ini, ini salah karena sistem sudah memasukkan layout yang dimekarkan ke dalam container—meneruskan true akan membuat tampilan grup berlebih dalam layout akhir.) Kita kini telah melihat cara membuat fragmen yang menyediakan layout. Berikutnya, kita perlu menambahkan fragmen ke aktivitas.

2.5 Menambahkan fragmen ke aktivitas

Fragmen berkontribusi pada sebagian UI ke aktivitas host, yang disematkan sebagai bagian dari hierarki tampilan keseluruhan aktivitas. Ada dua cara untuk menambahkan fragmen ke layout aktivitas:

1. Deklarasikan fragment dalam file layout aktivitas

Atribut android:name dalam <fragment> menetapkan class Fragment untuk dibuat instance-nya dalam layout.

Saat sistem membuat layout aktivitas, sistem membuat instance setiap fragmen sebagaimana yang ditetapkan dalam layout dan memanggil metode onCreateView() masing-masing, untuk mengambil setiap fragmen. Sistem akan menyisipkan View yang dikembalikan oleh fragmen secara langsung, menggantikan elemen <fragment>. Atau,

secara programatis tambahkan fragmen ke ViewGroup yang ada. Kapan saja saat aktivitas Atribut android:name dalam <fragment> menetapkan class Fragment untuk dibuat instance-nya dalam layout.

Saat sistem membuat layout aktivitas, sistem membuat instance setiap fragmen sebagaimana yang ditetapkan dalam layout dan memanggil metode onCreateView() masing-masing, untuk mengambil setiap fragmen. Sistem akan menyisipkan View yang dikembalikan oleh fragmen secara langsung, menggantikan elemen <fragment>. Atau, secara programatis tambahkan fragmen ke ViewGroup yang ada. Kapan saja saat aktivitas berjalan, Kita bisa menambahkan fragmen ke layout aktivitas. Kita cukup menetapkan ViewGroup di tempat memasukkan fragmen.

Untuk membuat transaksi fragmen dalam aktivitas (seperti menambah, membuang, atau mengganti fragmen), kita harus menggunakan API dari FragmentTransaction. Kita bisa mengambil instance FragmentTransaction dari FragmentActivity seperti ini:

```
val fragmentManager = supportFragmentManager
val fragmentTransaction = fragmentManager.beginTransaction()
```

Selanjutnya kita bisa menambahkan fragmen menggunakan metode add(), dengan menetapkan fragmen yang akan ditambahkan dan tampilan tempat menyisipkannya. Sebagai contoh:

```
val fragment = ExampleFragment()
fragmentTransaction.add(R.id.fragment_container, fragment)
fragmentTransaction.commit()
```

Argumen pertama yang diteruskan ke add() adalah ViewGroup tempat fragmen harus dimasukkan, yang ditetapkan oleh ID resource, dan parameter kedua merupakan fragmen yang akan ditambahkan. Setelah membuat perubahan dengan FragmentTransaction, Kita harus memanggil commit() untuk menerapkan perubahan.

2.6 Mengelola Fragmen

Untuk mengelola fragmen dalam aktivitas, kita perlu menggunakan `FragmentManager`. Untuk mendapatkannya, panggil `getSupportFragmentManager()` dari aktivitas kita.

Beberapa hal yang dapat Kita lakukan dengan `FragmentManager` antara lain:

1. Dapatkan fragmen yang ada di aktivitas dengan `findFragmentById()` (untuk fragmen yang menyediakan UI dalam layout aktivitas) atau `findFragmentByTag()` (untuk fragmen yang menyediakan atau tidak menyediakan UI).
2. Tarik fragmen dari back-stack, dengan `popBackStack()` (menyimulasikan perintah Kembali oleh pengguna).
3. Daftarkan listener untuk perubahan pada back-stack, dengan `addOnBackStackChangeListener()`.

2.7 Melakukan Transaksi Fragmen

Fitur menarik terkait penggunaan fragmen di aktivitas adalah kemampuan menambah, membuang, mengganti, dan melakukan tindakan lain dengannya, sebagai respons atas interaksi pengguna. Setiap set perubahan yang kita lakukan untuk aktivitas disebut transaksi dan kita bisa melakukan transaksi menggunakan API di `FragmentManager`. Kita juga bisa menyimpan setiap transaksi ke back-stack yang dikelola aktivitas, sehingga pengguna bisa mengarah mundur melalui perubahan fragmen (mirip mengarah mundur melalui aktivitas).

Kita bisa memperoleh instance `FragmentManager` dari `FragmentManager` seperti ini:

```
val fragmentManager = supportFragmentManager
val fragmentTransaction
```

```
= fragmentManager.beginTransaction()
```

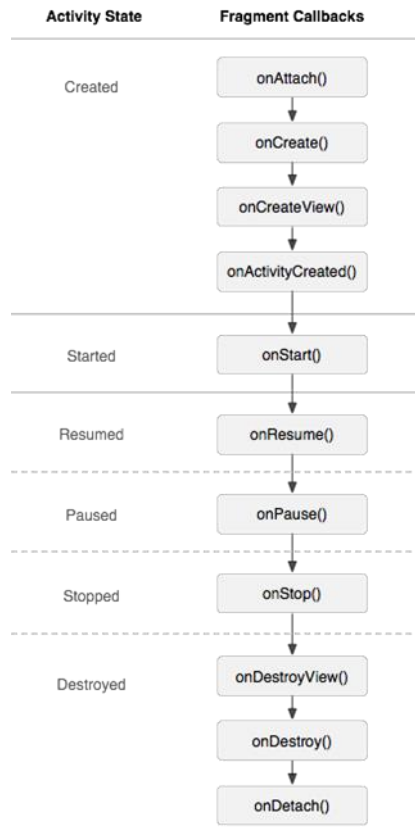
Setiap transaksi merupakan serangkaian perubahan yang ingin dilakukan pada waktu yang sama. Kita bisa menyiapkan semua perubahan yang ingin dilakukan untuk transaksi mana saja menggunakan metode seperti `add()`, `remove()`, dan `replace()`. Kemudian, untuk menerapkan transaksi pada aktivitas, kita harus memanggil `commit()`. Akan tetapi, sebelum memanggil `commit()`, kita mungkin perlu memanggil `addToBackStack()`, untuk menambahkan transaksi ke back-stack transaksi fragmen. Back-stack ini dikelola oleh aktivitas dan memungkinkan

pengguna kembali ke status fragmen sebelumnya, dengan menekan tombol Kembali. Misalnya, dengan cara ini kita bisa mengganti satu fragmen dengan yang fragmen lain, dan mempertahankan status sebelumnya di back-stack:

2.8 Menambahkan item ke Bilah Aplikasi

Fragmen kita bisa menyumbangkan item menu ke Menu Opsi aktivitas (dan, konsekuensinya, bilah aplikasi) dengan mengimplementasikan `onCreateOptionsMenu()`. Agar metode ini bisa menerima panggilan, Kita harus memanggil `setHasOptionsMenu()` selama `onCreate()`, untuk menunjukkan bahwa fragmen ingin menambahkan item ke Menu Opsi. Jika tidak, fragmen tidak menerima panggilan ke `onCreateOptionsMenu()`. Setiap item yang selanjutnya Kita tambahkan ke Menu Opsi dari fragmen akan ditambahkan ke item menu yang ada. Fragmen juga menerima callback ke `onOptionsItemSelected()` bila item menu dipilih. Kita juga bisa mendaftarkan tampilan dalam layout fragmen untuk menyediakan menu konteks dengan memanggil `registerForContextMenu()`. Bila pengguna membuka menu konteks, fragmen akan menerima panggilan ke `onCreateContextMenu()`. Bila pengguna memilih item, fragmen akan menerima panggilan ke `onContextItemSelected()`.

2.9 Menangani Daur Hidup Fragmen



Gambar 3. Efek daur hidup aktivitas pada daur hidup fragmen.

Mengelola daur hidup fragmen mirip sekali dengan mengelola daur hidup aktivitas. Seperti aktivitas, fragmen bisa berada dalam tiga status:

Dilanjutkan : Fragmen terlihat dalam aktivitas yang berjalan.

Dihentikan sementara : Aktivitas lain berada di latar depan dan memiliki fokus, namun aktivitas tempat fragmen berada masih terlihat (aktivitas latar depan sebagian terlihat atau tidak menutupi seluruh layar). Dihentikan

2.10 Mengoordinasi dengan daur hidup aktivitas

Daur hidup aktivitas tempat fragmen berada akan memengaruhi secara langsung siklus hidup fragmen sedemikian rupa sehingga setiap callback daur hidup aktivitas menghasilkan callback yang sama untuk masing-masing fragmen. Misalnya, bila aktivitas menerima dalam aktivitas akanonPause().

Namun fragmen memiliki beberapa callback menerima daur hidup ekstra, yang menangani interaksi unik dengan aktivitas untuk melakukan tindakan seperti membangun onPause(), maka masing-masing fragmen dan memusnahkan UI fragmen. Metode callback tambahan ini adalah:

onAttach dipanggil bila fragmen telah dikaitkan dengan aktivitas (Activity diteruskan di Dipanggil untuk membuat hierarki tampilan yang dikaitkan dengan fragmen. onActivityCreated() Dipanggil bila metode onCreate() aktivitas telah dikembalikan. onDestroyView()Dipanggil bila hierarki tampilan yang terkait dengan fragmen dihapus. onDetach()Dipanggil bila fragmen diputuskan dari aktivitas. Alur daur hidup fragmen, karena dipengaruhi oleh aktivitas host-nya, diilustrasikan oleh gambar 3. Dalam gambar tersebut, kita bisa melihat bagaimana setiap status aktivitas yang berurutan menentukan metode callback mana yang mungkin diterima fragmen. Misalnya, saat aktivitas menerima callback onCreate(), fragmen dalam aktivitas akan menerima tidak lebih dari callback onActivityCreated(). Setelah status aktivitas diteruskan kembali, Kita bisa bebas menambah dan membuang fragmen untuk aktivitas tersebut. Sehingga, hanya saat aktivitas berada dalam status dilanjutkan, daur hidup fragmen bisa berubah secara independen. Akan tetapi, saat aktivitas meninggalkan status dilanjutkan, fragmen akan kembali didorong melalui daur hidupnya oleh aktivitas.

BAB III TUGAS PENDAHULUAN

3.1 Soal

1. Apa itu fragment ?
2. Jelaskan dengan bahasa anda sendiri cara membuat fragment di kotlin !
3. Jelaskan perbedaan antara fragment dan activity di kotlin !
4. Bagaimana cara berkomunikasi antara fragment dengan activity kotlin
5. Sebutkan yang anda ketahui tentang back stack di fragment kotlin !

3.2 Jawaban

1. Fragment adalah bagian UI aplikasi yang dapat digunakan kembali. Sebuah fragment menentukan dan mengelola tata letaknya sendiri, memiliki siklus proses sendiri, serta dapat menangani proses inputnya sendiri. Fragment tidak dapat berjalan sendiri, fragment harus dihosting oleh aktivitas lain.
2. Cara membuat fragment di kotlin
Pertama, buat sebuah kotlin baru yang merupakan subclass dari "fragment". Selanjutnya didalam kelas "MyFragment" implementasikan data method utama yang biasanya digunakan dalam sebuah fragment. "onCreateView()" method ini dipanggil ketika fragment dibuat. Didalam method ini, perlu mengatur tampilan fragment dengan mengembalikan tampilan yang di inflate menggunakan "LayoutInflate".
Setelah membuat fragment, dapat menggunakan "FragmentManager" untuk menambahkannya ke aktivitas.
3. Perbedaan antara fragment dan activity di kotlin
Fragment adalah bagian modular dari antarmuka pengguna didalam sebuah activity fragment tidak berdiri sendiri dan harus melekat pada sebuah activity, beberapa fragment dapat digunakan bersama dalam satu activity untuk membuat antarmuka yang lebih kompleks dan fleksibel.

BAB IV

IMPLEMENTASI

4.1 Source Code

4.1.1 AndroidManifest

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

    <application
        android:allowBackup="true"

        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/Theme.PRAKTIKUM5"
        tools:targetApi="31">
        <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>
```

4.1.2 ChatFragment

```
package com.example.praktikum5

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

class ChatFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_chat,
            container, false)
    }
}
```

4.1.3 FragmentAdapter

```
package com.example.praktikum5

import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentManager
```

```

import androidx.fragment.app.FragmentStatePagerAdapter

class FragmentAdapter(fm : FragmentManager) :
    FragmentStatePagerAdapter (fm,
        BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT) {

    var fragmentList : ArrayList<Fragment> = ArrayList()
    var fragmentTitle : ArrayList<String> = ArrayList()

    override fun getCount(): Int {
        return fragmentList.size
    }

    override fun getItem(position: Int): Fragment {
        return fragmentList[position]
    }

    override fun getPageTitle(position: Int): CharSequence? {
        return fragmentTitle[position]
    }

    fun addFragment(fragment: Fragment, title : String){
        fragmentList.add(fragment)
        fragmentTitle.add(title)
    }
}

```

4.1.4 HomeFragment

```

package com.example.praktikum5

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

class HomeFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home,
            container, false)
    }
}

```

4.1.5 MainActivity

```

package com.example.praktikum5

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import androidx.viewpager.widget.ViewPager
import com.google.android.material.tabs.TabLayout

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

        var viewPager : ViewPager =

```



```

findViewById(R.id.viewPager) as ViewPager
    var tablayout : TabLayout =
findViewById(R.id.tablayout) as TabLayout

    val fragmentAdapter =
FragmentAdapter(supportFragmentManager)
    fragmentAdapter.addFragment(ChatFragment(), "Chat")
    fragmentAdapter.addFragment(HomeFragment(), "Status")
    fragmentAdapter.addFragment(SettingsFragment(),
"Settings")

    viewPager.adapter = fragmentAdapter
    tablayout.setupWithViewPager(viewPager)
}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    getMenuInflater().inflate(R.menu.action_menu, menu);
    return true;
}
}

```

4.1.6 SettingsFragment

```

package com.example.praktikum5

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

class SettingsFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_settings,
container, false)
    }
}

```

4.1.7 activity_main

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"
android:orientation="vertical"
tools:context=".MainActivity">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tablayout"
        android:background="@color/cadet"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabIndicatorColor="@color/black"
        app:tabSelectedTextColor="@color/black"
        app:tabTextAppearance="@style/TabLayoutTextStyle"
        app:tabTextColor="@color/black"/>

```

```

<androidx.viewpager.widget.ViewPager
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:id="@+id/viewPager"
    android:background="@color/white"
    android:layout_weight="1"/>
</LinearLayout>

```

4.1.8 fragment_chat

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ChatFragment">

    <!-- TODO: Update blank fragment layout -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:gravity="center_vertical">
            <de.hdodenhof.circleimageview.CircleImageView
                android:layout_marginLeft="10dp"
                android:src="@drawable/people"
                android:layout_width="50dp"
                android:layout_height="50dp">
            </de.hdodenhof.circleimageview.CircleImageView>
            <LinearLayout
                android:layout_width="250dp"
                android:layout_height="match_parent"
                android:gravity="center_vertical"
                android:orientation="vertical">
                <TextView
                    android:layout_marginLeft="20dp"
                    android:textStyle="bold"
                    android:textColor="@color/black"
                    android:text="Septiya"
                    android:layout_width="280dp"
                    android:layout_height="25dp">
                </TextView>
                <TextView
                    android:layout_marginLeft="20dp"
                    android:textSize="10dp"
                    android:text="Semangat Kuliah!!!"
                    android:layout_width="280dp"
                    android:layout_height="25dp">
                </TextView>
            </LinearLayout>
            <LinearLayout
                android:layout_width="20dp"
                android:layout_height="match_parent"
                android:gravity="center_vertical">
            </LinearLayout>
            <LinearLayout
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:gravity="center_vertical">
                <TextView
                    android:layout_marginLeft="20dp"
                    android:textSize="10dp"
                    android:text="00.00"
                    android:layout_width="60dp"

```

```

        android:layout_height="30dp">
    </TextView>
</LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="20dp"
        android:background="@drawable/archive"
        android:layout_width="30dp"
        android:layout_height="30dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">
    <TextView
        android:layout_marginLeft="20dp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:text="Chat Diarsipkan"
        android:layout_width="280dp"
        android:layout_height="25dp">
    </TextView>
</LinearLayout>
<LinearLayout
    android:layout_width="50dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical">
    <TextView
        android:layout_marginLeft="20dp"
        android:textSize="10dp"
        android:text="10"
        android:textColor="@color/black"
        android:layout_width="60dp"
        android:layout_height="30dp">
    </TextView>
</LinearLayout>

</LinearLayout>
</LinearLayout>
</FrameLayout>

```

4.1.9 fragment_home

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".HomeFragment">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:gravity="center_vertical">

```

```

<de.hdodenhof.circleimageview.CircleImageView
    android:layout_marginLeft="10dp"
    android:src="@drawable/people"
    android:layout_width="50dp"
    android:layout_height="50dp">
</de.hdodenhof.circleimageview.CircleImageView>
<LinearLayout
<LinearLayout
    android:layout_width="250dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">
    <TextView
        android:layout_marginLeft="20dp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:text="Status saya"
        android:layout_width="280dp"
        android:layout_height="25dp">
    </TextView>
    <TextView
        android:layout_marginLeft="20dp"
        android:textSize="10dp"
        android:text="10 menit yang lalu"
        android:layout_width="280dp"
        android:layout_height="25dp">
    </TextView>
</LinearLayout>
<LinearLayout
    android:layout_width="100dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical">
    <ImageButton
        android:layout_marginLeft="30dp"
        android:background="@drawable/more"
        android:layout_width="30dp"
        android:layout_height="30dp">
    </ImageButton>
</LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="30dp">
    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:gravity="center_vertical"
        android:orientation="vertical">
        <TextView
            android:layout_marginLeft="20dp"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:text="Pembaruan Status"
            android:layout_width="280dp"
            android:layout_height="25dp">
        </TextView>
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:gravity="center_vertical">
        <de.hdodenhof.circleimageview.CircleImageView
            android:layout_marginLeft="20dp"
            android:src="@drawable/people"
            android:layout_width="50dp"
            android:layout_height="50dp">

```

```

</de.hdodenhof.circleimageview.CircleImageView>
<LinearLayout
    android:layout_width="250dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">
    <TextView
        android:layout_marginLeft="20dp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:text="Niken"
        android:layout_width="280dp"
        android:layout_height="25dp">
    </TextView>
    <TextView
        android:layout_marginLeft="20dp"
        android:textSize="10dp"
        android:text="10 menit yang lalu"
        android:layout_width="280dp"
        android:layout_height="25dp">
    </TextView>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</FrameLayout>

```

4.1.10 Fragment_settings

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SettingsFragment">

    <!-- TODO: Update blank fragment layout -->
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="120dp">
            <de.hdodenhof.circleimageview.CircleImageView
                android:src="@drawable/people"
                android:padding="15dp"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            </de.hdodenhof.circleimageview.CircleImageView>
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="60dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:gravity="center_vertical"
                android:orientation="vertical">
                <TextView
                    android:layout_marginLeft="170dp"
                    android:textStyle="bold"
                    android:textColor="@color/black"
                    android:textSize="20dp"

```

```

        android:text="Septiya"
        android:layout_width="100dp"
        android:layout_height="25dp">
    </TextView>
</LinearLayout>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:padding="5dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="10dp"
        android:background="@drawable/supervisor"
        android:layout_width="30dp"
        android:layout_height="30dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"

android:orientation="vertical"></LinearLayout>
    </LinearLayout>
    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:gravity="center_vertical"
        android:orientation="vertical">
        <TextView
            android:layout_marginLeft="20dp"
            android:textColor="@color/black"
            android:text="Akun"
            android:layout_width="280dp"
            android:layout_height="20dp">
        </TextView>
        <TextView
            android:layout_marginLeft="20dp"
            android:textSize="10dp"
            android:text="Ganti Nama, Ganti Nomor"
            android:layout_width="280dp"
            android:layout_height="20dp">
        </TextView>
    </LinearLayout>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:padding="5dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="10dp"
        android:background="@drawable/lock"
        android:layout_width="30dp"
        android:layout_height="30dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"

android:orientation="vertical"></LinearLayout>
    </LinearLayout>
    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:gravity="center_vertical"
        android:orientation="vertical">
        <TextView
            android:layout_marginLeft="20dp"
            android:textColor="@color/black"

```

```

        android:text="Privasi"
        android:layout_width="280dp"
        android:layout_height="20dp">
    </TextView>
    <TextView
        android:layout_marginLeft="20dp"
        android:textSize="10dp"
        android:text="Blokir kontak, Pesan
sementara"
        android:layout_width="280dp"
        android:layout_height="20dp">
    </TextView>
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:padding="5dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="10dp"
        android:background="@drawable/notification"
        android:layout_width="30dp"
        android:layout_height="30dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">
    <TextView
        android:layout_marginLeft="20dp"
        android:textColor="@color/black"
        android:text="Notifikasi"
        android:layout_width="280dp"
        android:layout_height="20dp">
    </TextView>
    <TextView
        android:layout_marginLeft="20dp"
        android:textSize="10dp"
        android:text="Pesan, Grup, Nada Dering"
        android:layout_width="280dp"
        android:layout_height="20dp">
    </TextView>
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:padding="5dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="10dp"
        android:background="@drawable/chat"
        android:layout_width="30dp"
        android:layout_height="30dp">
    </LinearLayout>
    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:gravity="center_vertical"

```

```

        android:orientation="vertical">
        <TextView
            android:layout_marginLeft="20dp"
            android:textColor="@color/black"
            android:text="Chat"
            android:layout_width="280dp"
            android:layout_height="20dp">
        </TextView>
        <TextView
            android:layout_marginLeft="20dp"
            android:textSize="10dp"
            android:text="Tema, Walpaper, Riwayat chat"
            android:layout_width="280dp"
            android:layout_height="20dp">
        </TextView>
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:padding="5dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="10dp"
        android:background="@drawable/bahasa"
        android:layout_width="30dp"
        android:layout_height="30dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            </LinearLayout>
        </LinearLayout>
        <LinearLayout
            android:layout_width="300dp"
            android:layout_height="match_parent"
            android:gravity="center_vertical"
            android:orientation="vertical">
            <TextView
                android:layout_marginLeft="20dp"
                android:textColor="@color/black"
                android:text="Bahasa Aplikasi"
                android:layout_width="280dp"
                android:layout_height="20dp">
            </TextView>
            <TextView
                android:layout_marginLeft="20dp"
                android:textSize="10dp"
                android:text="Bahasa Indonesia (bahasa
telepon)"
                android:layout_width="280dp"
                android:layout_height="20dp">
            </TextView>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:padding="5dp"
    android:gravity="center_vertical">
    <LinearLayout
        android:layout_marginLeft="10dp"
        android:background="@drawable/outline"
        android:layout_width="30dp"
        android:layout_height="30dp">

        <LinearLayout

```



```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">
    <TextView
        android:layout_marginLeft="20dp"
        android:textColor="@color/black"
        android:text="Bantuan"
        android:layout_width="280dp"
        android:layout_height="20dp">
    </TextView>
    <TextView
        android:layout_marginLeft="20dp"
        android:textSize="10dp"
        android:text="Pusat bantuan, hubungi kami,
kebiakan privasi"
        android:layout_width="280dp"
        android:layout_height="20dp">
    </TextView>
</LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:gravity="center_horizontal|center">
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:gravity="center_horizontal|center">
    <TextView
        android:layout_marginLeft="70dp"
        android:textSize="10dp"
        android:text="from -SY-"
        android:layout_width="100dp"
        android:layout_height="25dp">
    </TextView>
</LinearLayout>
</LinearLayout>
</FrameLayout>

```

4.1.11 Action_menu

```

<?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/menu_status"
        android:icon="@drawable/camera"
        android:title="Kamera"
        app:showAsAction="always"/>
    <item
        android:id="@+id/menu_search"
        android:icon="@drawable/search"
        android:title="Pencarian"
        app:showAsAction="always"/>
    <item
        android:id="@+id/menu_grub"
        android:title="Grub Baru"
        app:showAsAction="never"/>

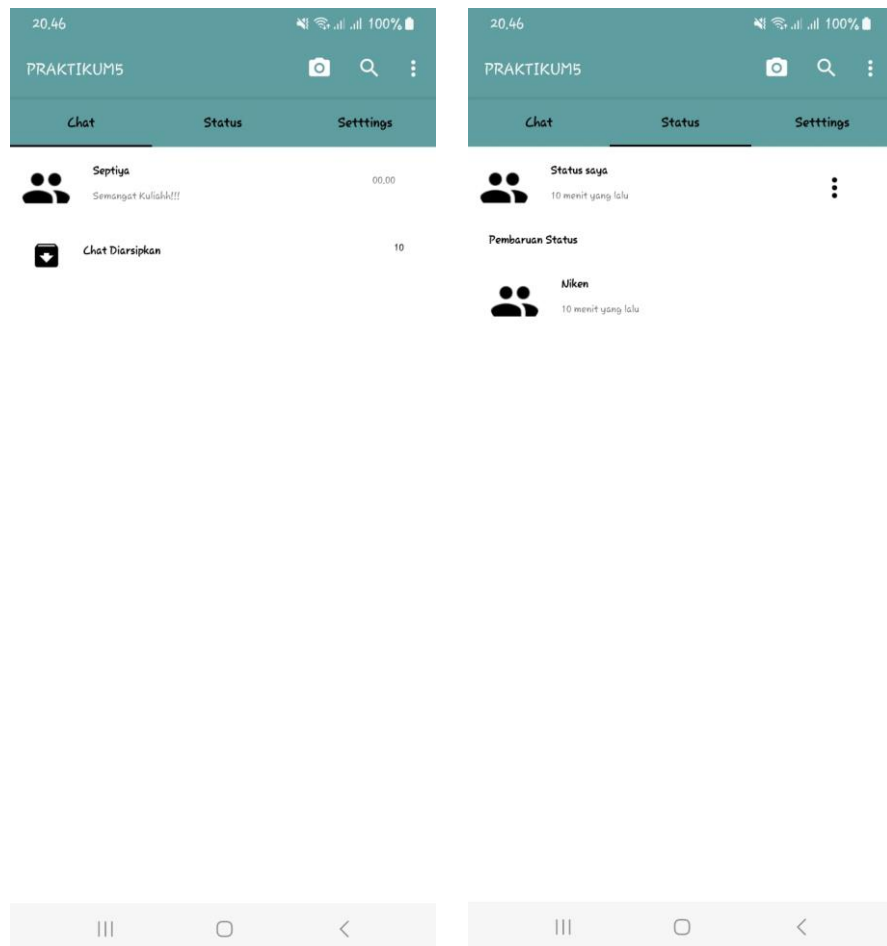
```

```

<item
    android:id="@+id/menu_siaran"
    android:title="Siaran Baru"
    app:showAsAction="never"/>
<item
    android:id="@+id/menu_perangkat"
    android:title="Perangkat Tertaut"
    app:showAsAction="never"/>
<item
    android:id="@+id/menu_pesan"
    android:title="Pesan Berbintang"
    app:showAsAction="never"/>
<item
    android:id="@+id/menu_pengaturan"
    android:title="Setelan"
    app:showAsAction="never"/>
</menu>

```

4.2 Hasil Run





BAB V

PENUTUP

5.1 Analisa

Modul yang mempelajari tentang Fragment 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 sebuah halaman baru di dalam aplikasi Android menggunakan Android Studio. Fragment menghadirkan modularitas dan kemampuan penggunaan kembali ke dalam UI aktivitas dengan memungkinkan membagi UI menjadi bagian yang terpisah. Aktivitas merupakan tempat ideal untuk menempatkan elemen global di sekitar antarmuka pengguna aplikasi, seperti panel navigasi. Sebaliknya, fragmen lebih cocok untuk menentukan dan mengelola UI dari satu atau sebagian layar. Membagi UI menjadi beberapa fragmen akan memudahkan kita dalam mengubah tampilan aktivitas saat runtime. Kita dapat menggunakan beberapa instance dari class fragmen yang sama dalam aktivitas yang sama, di beberapa aktivitas, atau bahkan sebagai turunan dari fragmen lain. Secara keseluruhan, modul ini akan memberikan pengguna pemahaman yang lebih mendalam tentang cara memanfaatkan fragment untuk meningkatkan kualitas dan fungsionalitas aplikasi Android yang dibuat.

5.2 Kesimpulan

Dari modul 5 ini kami dapat mempelajari Fragment di Android Studio, dan dapat disimpulkan bahwasannya bagian UI aplikasi Anda yang dapat digunakan kembali. Sebuah fragmen menentukan dan mengelola tata letaknya sendiri, memiliki siklus proses sendiri, serta dapat menangani peristiwa inputnya sendiri. Fragment tidak dapat berjalan sendiri. Fragment harus dihosting oleh aktivitas atau fragmen lain. Hierarki tampilan fragmen menjadi bagian dari, atau dilampirkan ke, hierarki tampilan host. Dengan memahami materi tentang fragment 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.