

LAPORAN PRAKTIKUM PENGEMBANGAN APLIKASI BERGERAK

Android Fundamental 1 - WEEK 6



Disusun oleh:

Nama : Afif Imam Rahadi

Nim : L0122006

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

1. Screenshot Source Code

Ini adalah screenshot dan juga sedikit penjelasan dari setiap halaman yang sudah saya buat dalam aplikasi Line Lite ini. Terdapat halaman menu, chats, news, dan call. Pada halaman menu terdapat pengiriman data menggunakan bundle yang nantinya akan menuju halaman bundle fragment.

A. Main

```
package com.l0122006.afifimam.linelite

import ...

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

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

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val navView: BottomNavigationView = binding.navView

        val navController = findNavController(R.id.nav_host_fragment_activity_main)
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        val appBarConfiguration = AppBarConfiguration(
            listOf(
                R.id.navigation_home,
                R.id.navigation_chats,
                R.id.navigation_calls,
                R.id.navigation_news
            )
        )
        setupActionBarWithNavController(navController, appBarConfiguration)
        navView.setupWithNavController(navController)
    }
}
```

Pada source code mainActivity.kt dimulai dengan melakukan impor-import yang diperlukan karena nantinya akan digunakan dalam program. Ini termasuk impor untuk kelas-kelas seperti AppCompatActivity, Bundle, dan elemen-elemen yang terkait dengan tata letak navigasi. Selanjutnya, terdapat class MainActivity dideklarasikan yang mewarisi AppCompatActivity. Pada method onCreate, layout utama di-inflate menggunakan ActivityMainBinding, yang merupakan file binding berdasarkan layout XML activity_main.xml.

Konfigurasi BottomNavigationView diambil dari binding.NavController diinisialisasi dengan menemukan host fragment untuk navigasi, yang diidentifikasi oleh ID nav_host_fragment_activity_main. AppBarConfiguration dibuat dengan menetapkan set dari ID-nya masing-masing tampilan utama yang akan ditampilkan di bottom navigation. setupActionBarWithNavController() digunakan untuk menghubungkan ActionBar dengan NavController, sehingga judul ActionBar akan berubah sesuai dengan tampilan yang sedang aktif.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.af
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />

    <fragment
        android:id="@+id/nav_host_fragment_activity_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

```

Source code ctivity_main.xml menggunakan bottom navigation. Layout ini menggunakan ConstraintLayout. Ini adalah root layout yang digunakan untuk mengatur tata letak elemen-elemen dalam tampilan. Selanjutnya terdapat container, ini adalah ID dari ConstraintLayout yang nantinya digunakan untuk merujuk ke layout utama dalam kode program. Berikutnya, BottomNavigationView yang merupakan komponen UI untuk bottom navigation. Itu ditempatkan di bagian bawah layar dan memiliki tautan ke menu navigasi (@menu/bottom_nav_menu). Layout ini diatur melalui layout_width diatur menjadi "0dp" agar bisa menggunakan ConstraintLayout secara efektif. Tidak hanya itu, layout_constraintBottom_toBottomOf="parent" menempatkannya di bagian bawah layout. Berikutnya NavHostFragment, ini adalah fragment yang bertindak sebagai host untuk semua tujuan navigasi. Ini memanfaatkan NavController untuk menavigasi antara tujuan-tujuan. defaultNavHost diatur sebagai "true" sehingga NavController dapat menangani tombol kembali secara tepat. Terakhir, app:navGraph="@navigation/mobile_navigation" merujuk pada file navigasi (mobile_navigation.xml) yang berisi definisi tujuan-tujuan dan jalur navigasi antara mereka.

B. Navigation

```
<?xml version="1.0" encoding="utf-8"?>
<navigation 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:id="@+id/mobile_navigation"
    app:startDestination="@+id/navigation_home">

    <fragment
        android:id="@+id/navigation_home"
        android:name="com.l0122006.afifimam.linelite.ui.home.HomeFragment"
        android:label="Home"
        tools:layout="@layout/fragment_home" >
        <action
            android:id="@+id/action_navigation_home_to_bundleFragment2"
            app:destination="@id/bundleFragment" />
        </fragment>

    <fragment
        android:id="@+id/navigation_chats"
        android:name="com.l0122006.afifimam.linelite.ui.chats.ChatsFragment"
        android:label="Chats"
        tools:layout="@layout/fragment_chats" />

    <fragment
        android:id="@+id/navigation_calls"
        android:name="com.l0122006.afifimam.linelite.ui.calls.CallsFragment"
        android:label="Calls"
        tools:layout="@layout/fragment_calls" />

</navigation>
```

Source code mobile_navigation menggunakan navigation sebagai root utaman. Selanjutnya terdapat android:id="@+id/mobile_navigation" Ini adalah ID unik untuk file navigasi ini. Berikutnya, app:startDestination="@+id/navigation_home": Ini menentukan tujuan awal yang akan ditampilkan saat navigasi dimulai. Di dalam source code ini terdapat Fragment dan setiap elemen fragment mewakili satu tujuan dalam aplikasi. android:id: Ini adalah ID unik untuk tujuan tersebut. android:name ini adalah nama kelas Java/Kotlin untuk fragment yang akan ditampilkan. android:label adalah label yang akan digunakan untuk tujuan tersebut. Action adalah tindakan yang menghubungkan dua tujuan. Dalam kasus ini, ada satu tindakan yang menghubungkan navigation_home dengan bundleFragment.

```

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

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/baseline_apps_24"
        android:title="Home" />

    <item
        android:id="@+id/navigation_chats"
        android:icon="@drawable/baseline_attach_email_24"
        android:title="Chats" />

    <item
        android:id="@+id/navigation_news"
        android:icon="@drawable/baseline_announcement_24"
        android:title="News" />

    <item
        android:id="@+id/navigation_calls"
        android:icon="@drawable/baseline_add_call_24"
        android:title="Calls" />

</menu>

```

Source code `bottom_nav_menu` menggunakan `menu` sebagai root struktur menu. `xmlns:android="http://schemas.android.com/apk/res/android"`: Ini adalah namespace untuk elemen-elemen Android. Item merupakan setiap elemen item mewakili satu item dalam menu bottom navigation. `android:id` adalah ID unik untuk item tersebut, yang akan digunakan untuk mengidentifikasi item saat dipilih. `android:icon` adalah ikon yang akan ditampilkan di sebelah teks item yang dapat diambil sourcenya pada folder `drawable`.

C. Home

```

package com.l0122006.afifimam.linelite.ui.home

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.navigation.Navigation
import com.l0122006.afifimam.linelite.R
import com.l0122006.afifimam.linelite.databinding.FragmentHomeBinding

class HomeFragment : Fragment() {

    private var _binding: FragmentHomeBinding? = null
    private val binding get() = _binding!!

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View {
        // Inflate the layout for this fragment
        _binding = FragmentHomeBinding.inflate(inflater, container, attachToParent: false)
        val view = binding.root
        return view
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        binding.btnCategory.setOnClickListener {
            Navigation.createNavigateOnClickListener(R.id.action_navigation_home_to_bundleFragment2)
        }
    }
}

```

Program dimulai dengan mendeklarasikan package dan mengimpor kelas-kelas yang diperlukan. Selanjutnya terdapat class yang merupakan bagian dari fragment untuk tampilan utama (Home) dalam aplikasi. Kelas ini merupakan turunan dari kelas Fragment. Selanjutnya, terdapat Dua properti privat (_binding dan binding) digunakan untuk mengakses binding yang digunakan untuk mengakses elemen UI dalam fragment. Metode onCreateView() digunakan untuk menginflate layout fragment menggunakan FragmentHomeBinding. Metode onViewCreated() dipanggil setelah tampilan fragment selesai dibuat. Di sini, listener pada tombol btnCategory ditetapkan untuk menavigasi ke tujuan lain (bundleFragment) menggunakan Navigation.createNavigateOnClickListener(). Metode onDestroy() digunakan untuk membersihkan binding saat fragment dihancurkan. Binding digunakan untuk mengakses elemen UI dalam layout fragment tanpa perlu menggunakan findViewById(). Ini membantu dalam membuat kode lebih bersih dan lebih aman.

```
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".ui.home.HomeFragment">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:padding="16dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical">

            <ImageView
                android:id="@+id/image_profile"
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:contentDescription="foto-profile"
                android:src="@drawable/profile"
                android:scaleType="fitCenter"/>
```

Xml ini menggunakan LinearLayout sebagai root layout. Pada bagian awal, terdapat CardView yang digunakan untuk menampilkan konten dalam bingkai. Penggunaan cardview untuk mengelompokkan gambar profil dan teks dalam satu kartu. Memiliki satu LinearLayout di dalamnya untuk menempatkan gambar profil dan teks. Tag yang dipakai antara lain ImageView untuk menampilkan gambar, TextView untuk menampilkan teks, dan Button untuk menampilkan tombol. Pada setiap tag, diawali dengan id untuk mengidentifikasi elemen ini. Terdapat source juga yang artinya mengambil gambar dari folder drawable. Terakhir, pada bagian bawah, terdapat button yang akan mengarah ke halaman fragment bundle.

D. Chats

```
package com.l0122006.afifimam.linelite.ui.chats

import ...

class ChatsFragment : Fragment() {

    private lateinit var rvChats: RecyclerView
    private val list = ArrayList<Chat>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val rootView = inflater.inflate(R.layout.fragment_chats, container, attachToRoot: false)

        rvChats = rootView.findViewById(R.id.rv_chats)
        rvChats.setHasFixedSize(true)

        list.addAll(getListChats())
        showRecyclerView()

        return rootView
    }

    private fun getListChats(): ArrayList<Chat> {
        val dataName = resources.getStringArray(R.array.data_name)
        val dataDesc = resources.getStringArray(R.array.data_desc_chats)
        val dataImg = resources.obtainTypedArray(R.array.data_img)
        val listChat = ArrayList<Chat>()
        for (i in dataName.indices) {
```

Program dimulai dengan mendeklarasikan package dan mengimpor kelas-kelas yang diperlukan. Selanjutnya, Class ChatsFragment yang merupakan bagian dari fragment untuk tampilan daftar obrolan (chats) dalam aplikasi. Kelas ini merupakan turunan dari kelas Fragment. Mempunyai properti rvChats untuk merepresentasikan RecyclerView yang digunakan untuk menampilkan daftar obrolan. Terdapat ArrayList list yang akan berisi data obrolan. Methods onCreateView() digunakan untuk menginflate layout fragment dan menginisialisasi RecyclerView serta menampilkan daftar obrolan. Method getListChats() digunakan untuk mendapatkan data obrolan dari sumber daya string dan array gambar, kemudian memasukkannya ke dalam ArrayList list. Method showRecyclerView() digunakan untuk menampilkan daftar obrolan dalam RecyclerView dengan menggunakan LinearLayoutManager.

```
package com.l0122006.afifimam.linelite

import ...

@Parcelize
data class Chat(
    val name: String,
    val desc: String,
    val img: Int
) : Parcelable
```

Seperti biasa program dimulai dengan mendeklarasikan package dan mengimpor kelas-kelas yang diperlukan. Selanjutnya, terdapat data class chat. @Parcelize digunakan untuk menandai bahwa kelas ini dapat di-parcelize yang memungkinkan objek dari kelas ini untuk dikirim antar komponen aplikasi (misalnya, antar aktivitas) melalui proses parceling, seperti saat melewati data antara aktivitas. Menggunakan data class karena secara otomatis akan menyediakan fungsi-fungsi bawaan. Properti-properti dari data class ini adalah name, desc, dan img. Kelas ini mengimplementasikan Parcelable, sehingga objek Chat dapat di-parcelize untuk pengiriman antar komponen aplikasi.

```
package com.l0122006.afifimam.linelite

import ...

class ListChatAdapter (private val listChat: ArrayList<Chat>) : RecyclerView.Adapter<ListChatAdapter.ViewHolder>() {
    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
        val tvName: TextView = itemView.findViewById(R.id.tv_item_name)
        val tvDescription: TextView = itemView.findViewById(R.id.tv_item_description)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val view: View = LayoutInflater.from(parent.context).inflate(R.layout.item_row_chat, parent, attachToRoot: false)
        return ViewHolder(view)
    }

    override fun getItemCount(): Int = listChat.size

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val (name, desc, img) = listChat[position]
        holder.imgPhoto.setImageResource(img)
        holder.tvName.text = name
        holder.tvDescription.text = desc
    }
}
```

Program dimulai dengan mendeklarasikan package dan mengimpor kelas-kelas yang diperlukan. Class ListChatAdapter merupakan adapter untuk RecyclerView. Class ini menerima sebuah ArrayList listChat sebagai parameter konstruktor yang berisi data obrolan yang akan ditampilkan. Di sini mengimplementasikan Class ViewHolder adalah kelas ViewHolder yang digunakan untuk merepresentasikan tampilan setiap item dalam RecyclerView. Di sini juga memiliki properti-properti untuk elemen-elemen tampilan dalam item, seperti gambar profil, nama, dan deskripsi obrolan. onCreateViewHolder() merupakan metode yang dipanggil ketika RecyclerView membutuhkan sebuah ViewHolder baru. Di sini, layout item untuk obrolan (item_row_chat.xml) diinflate dan ViewHolder baru dibuat dengan menggunakan layout tersebut. getItemCount() mengembalikan jumlah item dalam daftar obrolan. Jumlah item yang dikembalikan sesuai dengan panjang ArrayList listChat. onBindViewHolder() dipanggil ketika RecyclerView ingin menampilkan data pada posisi tertentu. Di sini, data obrolan pada posisi tertentu diambil dari ArrayList listChat, kemudian dihubungkan dengan ViewHolder yang sesuai. Data obrolan tersebut ditampilkan dalam elemen-elemen tampilan yang ada dalam ViewHolder.


```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android
xmlns:card_view="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/card_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginStart="8dp"
    android:layout_marginTop="4dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="4dp"
    card_view:cardCornerRadius="4dp">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp">

        <ImageView
            android:id="@+id/img_item_photo"
            android:layout_width="70dp"
            android:layout_height="100dp"
            android:scaleType="fitCenter"
            card_view:layout_constraintBottom_toBottomOf="parent"
            card_view:layout_constraintStart_toStartOf="parent"
            card_view:layout_constraintTop_toTopOf="parent"
            android:contentDescription="avatar"
            tools:src="@tools:sample/avatars" />

```

Program di atas adalah sebuah layout XML yang menampilkan tampilan item dalam daftar obrolan menggunakan CardView. Setiap item terdiri dari gambar profil, nama kontak, dan deskripsi kontak. Gambar profil ditampilkan di sebelah kiri dengan ukuran tertentu, sementara nama kontak dan deskripsi kontak ditampilkan di sebelah kanan. Layout ini menggunakan ConstraintLayout untuk menata elemen-elemen secara relatif, memastikan bahwa mereka diatur dengan benar dalam CardView. Selain itu, penggunaan atribut `layout_constraint` membuat tampilan item ini dapat disesuaikan dengan ukuran dan orientasi layar yang berbeda.

```

<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="10dp"
        android:gravity="center_vertical">

        <ImageView
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:src="@drawable/search"
            android:layout_marginEnd="8dp"
            android:layout_marginStart="10dp"
            android:contentDescription="Search Icon" />

        <EditText
            android:id="@+id/et_search"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:minWidth="100dp"

```

Source code ini menggunakan Linear Layout. LinearLayout horizontal digunakan untuk menempatkan ikon pencarian dan EditText untuk masukan pencarian. Memiliki ImageView untuk ikon pencarian (search) dan EditText untuk memasukkan teks pencarian. EditText memiliki ID et_search yang digunakan untuk mengidentifikasi elemen ini. Diberikan berat (layout_weight) sehingga EditText akan memanjang dan mengisi ruang yang tersedia. Selain itu, juga menggunakan Recycle View yang akan menampilkan daftar obrolan. Memiliki ID rv_chats yang digunakan untuk mengidentifikasi elemen ini. Diberikan layout_width dan layout_height yang sama dengan match_parent sehingga RecyclerView akan mengisi seluruh ruang yang tersedia dalam LinearLayout.

E. News

```
package com.l0122006.afifimam.linelite.ui.news

import ...

class NewsFragment : Fragment() {

    private lateinit var rvNews: RecyclerView
    private val list = ArrayList<News>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val rootView = inflater.inflate(R.layout.fragment_news, container, attachToRoot = false)

        rvNews = rootView.findViewById(R.id.rv_news)
        rvNews.setHasFixedSize(true)

        list.addAll(getListNews())
        showRecyclerView()

        return rootView
    }

    private fun getListNews(): ArrayList<News> {
        val dataName = resources.getStringArray(R.array.data_name_news)
        val dataDesc = resources.getStringArray(R.array.data_desc_news)
        val dataImg = resources.obtainTypedArray(R.array.data_img_news)
        val listNews = ArrayList<News>()
        for (i in dataName.indices) {
```

Program di atas adalah bagian dari sebuah aplikasi Android yang menampilkan daftar berita dalam sebuah fragment. Kelas NewsFragment merupakan turunan dari Fragment yang bertanggung jawab untuk menampilkan daftar berita dalam RecyclerView. Saat fragment ditampilkan, layout diinflate, RecyclerView diinisialisasi, dan data berita dimasukkan ke dalam ArrayList. Data berita kemudian ditampilkan dalam RecyclerView menggunakan adapter ListNewsAdapter, yang mengatur tampilan setiap item berita. Alur kerja program ini memastikan bahwa pengguna dapat melihat daftar berita dengan baik terstruktur dan dapat mengakses detail berita jika diperlukan.

```
package com.l0122006.afifimam.linelite

import ...

@Parcelize
data class News(
    val name: String,
    val desc: String,
    val img: Int
): Parcelable
```

Program di atas adalah sebuah data class News yang digunakan untuk merepresentasikan data berita. Data class ini memiliki tiga properti: name untuk judul berita, desc untuk deskripsi berita, dan img untuk sumber gambar berita. Anotasi @Parcelize digunakan untuk menandai bahwa kelas ini dapat di-parcelize, memungkinkan objek dari kelas ini untuk dikirim antar komponen aplikasi menggunakan proses parceling. Ini berguna saat melewati data antar aktivitas atau fragment dalam aplikasi Android. Selain itu, implementasi Parcelable memastikan objek News dapat dengan mudah diserialisasi dan deserialisasi.

```

package com.l0122006.afifimam.linelite

import ...

class ListNewsAdapter (private val listNews: ArrayList<News>): RecyclerView.Adapter<ListNewsAdapter.ListViewHolder>() {
    class ListViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){
        val imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
        val tvName: TextView = itemView.findViewById(R.id.tv_item_name)
        val tvDescription: TextView = itemView.findViewById(R.id.tv_item_description)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
        val view: View = LayoutInflater.from(parent.context).inflate(R.layout.item_row_news, parent, attachToRoot: false)
        return ListViewHolder(view)
    }

    override fun getItemCount(): Int = listNews.size

    override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
        val (name, desc, img) = listNews[position]
        holder.imgPhoto.setImageResource(img)
        holder.tvName.text = name
        holder.tvDescription.text = desc
    }
}

```

Program di atas adalah sebuah adapter RecyclerView dalam aplikasi Android yang disebut ListNewsAdapter. Adapter ini bertanggung jawab untuk menghubungkan data berita dengan tampilan item dalam RecyclerView. Kelas ini mengimplementasikan tiga metode utama dari RecyclerView.Adapter: onCreateViewHolder(), getItemCount(), dan onBindViewHolder(). Metode onCreateViewHolder() digunakan untuk membuat ViewHolder baru untuk setiap item dalam daftar berita, sementara getItemCount() mengembalikan jumlah total item dalam daftar. Metode onBindViewHolder() mengikat data berita ke ViewHolder sesuai dengan posisi dalam daftar, memastikan bahwa data tersebut ditampilkan dengan benar dalam tampilan item. Alur kerja program ini adalah mengatur tampilan item berita dalam RecyclerView dengan memanfaatkan ViewHolder yang telah dibuat.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/tv_item_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:layout_marginBottom="15dp"
        android:text="Title of the News" />

    <ImageView
        android:id="@+id/img_item_photo"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_marginBottom="20dp"
        android:scaleType="fitCenter"
        android:src="@drawable/kontak3"
        android:contentDescription="News Image" />

    <TextView
        android:id="@+id/tv_item_description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"

```

Program di atas adalah file XML layout yang digunakan untuk menampilkan tampilan detail berita. Layout ini terdiri dari beberapa elemen UI yang ditata secara vertikal dalam LinearLayout, termasuk judul berita, gambar berita, deskripsi berita, ikon komentar, jumlah komentar, ikon berbagi, dan jumlah berbagi. Elemen-elemen ini dirancang untuk memberikan informasi yang jelas dan mudah dibaca kepada pengguna tentang berita tertentu, serta memberikan opsi untuk berinteraksi dengan berita melalui komentar dan berbagi. Penggunaan atribut `layout_width` dan `layout_height` yang sesuai serta penataan yang baik memastikan tampilan berita tersebut terlihat baik dan mudah digunakan.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_news"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

Program di atas adalah file XML layout yang digunakan dalam aplikasi Android, yang menampilkan sebuah RecyclerView. RecyclerView ini bertanggung jawab untuk menampilkan daftar berita dalam sebuah activity MainActivity. Layout ini terdiri dari sebuah LinearLayout dengan orientasi vertikal yang mengisi seluruh ruang layar dengan RecyclerView. Penggunaan atribut `layout_width` dan `layout_height` yang sesuai memastikan bahwa RecyclerView menempati seluruh ruang yang tersedia dalam activity MainActivity. Ini memungkinkan pengguna untuk melihat daftar berita yang dapat di-scroll ke atas dan ke bawah saat digunakan dalam aplikasi.

F. Calls

```
package com.l0122006.afifimam.linelite.ui.calls

import ...

class CallsFragment : Fragment() {

    private lateinit var rvCalls: RecyclerView
    private val list = ArrayList<Call>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val rootView = inflater.inflate(R.layout.fragment_calls, container, attachToRoot: false)

        rvCalls = rootView.findViewById(R.id.rv_calls)
        rvCalls.setHasFixedSize(true)

        list.addAll(getListCalls())
        showRecyclerView()

        return rootView
    }

    private fun getListCalls(): ArrayList<Call> {
        val dataName = resources.getStringArray(R.array.data_name)
        val dataDate = resources.getStringArray(R.array.data_date_calls)
        val dataImg = resources.obtainTypedArray(R.array.data_img)
        val listCall = ArrayList<Call>()
        for (i in dataName.indices) {
            val call = Call(dataName[i], dataDate[i], dataImg.getResourceId(i, defValue: -1))
        }
    }
}
```

Program di atas adalah sebuah Fragment dalam aplikasi Android yang disebut CallsFragment. Fragment ini bertanggung jawab untuk menampilkan daftar panggilan telepon dalam sebuah RecyclerView. Saat Fragment ini dibuat, ia mengambil data panggilan telepon dari sumber daya string dan array gambar dalam file XML. Kemudian, data ini dimasukkan ke dalam ArrayList dan ditampilkan dalam RecyclerView menggunakan adapter yang sesuai. Ini memungkinkan pengguna untuk melihat daftar panggilan telepon yang dapat di-scroll ke atas dan ke bawah saat digunakan dalam aplikasi.

```
package com.l0122006.afifimam.linelite

import ...

@Parcelize
data class Call(
    val name: String,
    val desc: String,
    val img: Int
) : Parcelable
```

Program di atas merupakan bagian dari model data dalam aplikasi Android yang menggunakan Kotlin. Ini mendefinisikan sebuah data class bernama Call yang diimplementasikan sebagai parcelable. Data class ini memiliki tiga properti: name, desc, dan img, yang mewakili nama kontak, tanggal panggilan, dan gambar avatar untuk panggilan. Dengan menerapkan parcelable, objek Call dapat dengan mudah dikirimkan melalui intent antar komponen dalam aplikasi Android.

```
package com.l0122006.afifimam.linelite

import ...

class ListCallAdapter (private val listCall: ArrayList<Call>) : RecyclerView.Adapter<ListCallAdapter.ListViewHolder>() {
    class ListViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
        val tvName: TextView = itemView.findViewById(R.id.tv_item_name)
        val tvDescription: TextView = itemView.findViewById(R.id.tv_item_description)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
        val view: View = LayoutInflater.from(parent.context).inflate(R.layout.item_row_call, parent, attachToRoot: false)
        return ListViewHolder(view)
    }

    override fun getItemCount(): Int = listCall.size

    override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
        val (name, desc, img) = listCall[position]
        holder.imgPhoto.setImageResource(img)
        holder.tvName.text = name
        holder.tvDescription.text = desc
    }
}
```

Program di atas adalah sebuah adapter untuk RecyclerView. Adapter ini disebut ListCallAdapter dan bertanggung jawab untuk menghubungkan data panggilan telepon dengan tampilan item dalam RecyclerView. Setiap item dalam RecyclerView menampilkan informasi tentang panggilan, seperti nama kontak, tanggal panggilan, dan gambar avatar. Dengan menggunakan ViewHolder, adapter ini mengoptimalkan kinerja RecyclerView dengan mendaur ulang tampilan yang sudah ada dan hanya memperbarui data yang berubah saat diperlukan.

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_calls"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

Kode XML tersebut mendefinisikan tata letak utama untuk aktivitas MainActivity. Tata letak ini terdiri dari sebuah RecyclerView yang bertanggung jawab untuk menampilkan daftar panggilan telepon. RecyclerView diberi ID rv_calls dan ditetapkan agar memiliki lebar dan tinggi yang sesuai dengan tata letak utama. Saat aplikasi berjalan, RecyclerView akan diisi dengan item-item yang diproses oleh adapter yang sesuai.


```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/card_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginStart="8dp"
    android:layout_marginTop="4dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="4dp"
    card_view:cardCornerRadius="4dp">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp">

        <ImageView
            android:id="@+id/img_item_photo"
            android:layout_width="70dp"
            android:layout_height="100dp"
            android:scaleType="fitCenter"
            card_view:layout_constraintBottom_toBottomOf="parent"
            card_view:layout_constraintStart_toStartOf="parent"
            card_view:layout_constraintTop_toTopOf="parent"
            android:contentDescription="avatar"
            tools:src="@tools:sample/avatars" />

```

Kode XML tersebut mendefinisikan tampilan item untuk ditampilkan dalam RecyclerView. Item ini berupa CardView yang berisi gambar profil pengguna, ikon panggilan, nama kontak, dan deskripsi panggilan. Setiap item direpresentasikan sebagai ConstraintLayout yang menampilkan informasi tersebut dengan tata letak yang teratur, dengan gambar profil di bagian kiri dan ikon panggilan di bagian kanan.

G. Bundle

```
package com.l0122006.afifam.linelite

import ...

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [BundleFragment.newInstance] factory method to
 * create an instance of this fragment.
 */
class BundleFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let { it: Bundle
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }
    }

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

Kode tersebut merupakan sebuah Fragment yang dirancang untuk menerima dan menampilkan data yang dikirimkan melalui sebuah Bundle. Fragment ini memiliki dua parameter yang dapat diinisialisasi saat pembuatan instansinya. Melalui metode newInstance, pengguna dapat membuat sebuah instance baru dari BundleFragment dengan menyertakan nilai-nilai untuk kedua parameter tersebut. Saat onCreateView dipanggil, tata letak yang sesuai akan diinflate untuk fragment ini.

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:layout_marginBottom="15dp"
        android:text="Xavi Ikut Puji Taktik Bertahan Madrid saat Singkirkan

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_marginBottom="20dp"
        android:scaleType="fitCenter"
        android:src="@drawable/madrid"
        android:contentDescription="News Image" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textAlignment="center"
        android:layout_marginTop="5dp"
        android:text="Keberhasilan Real Madrid menyingkirkan Manchester City

</LinearLayout>

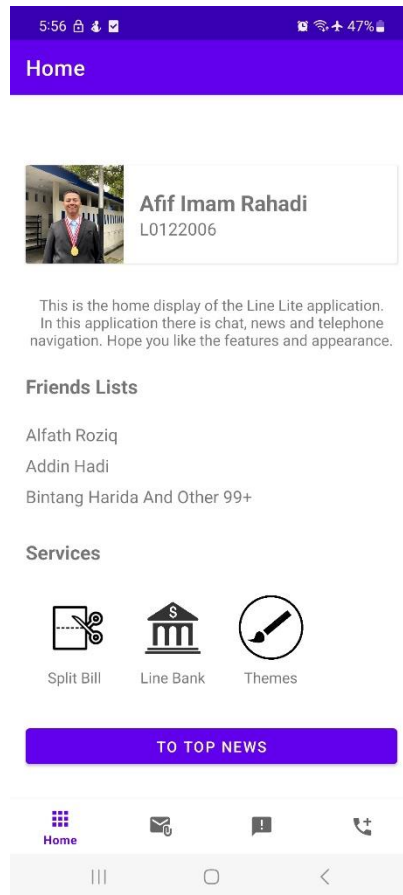
```

Kode tersebut merupakan tata letak XML untuk menampilkan konten berita pertama atau yang paling atas. Tata letak tersebut terdiri dari tiga elemen, yaitu sebuah TextView untuk judul berita, sebuah ImageView untuk gambar berita, dan sebuah TextView untuk isi berita. Semua elemen tersebut disusun dalam sebuah LinearLayout dengan orientasi vertikal dan memiliki padding 16dp di sekitar kontennya.

2. Screenshot Emulator

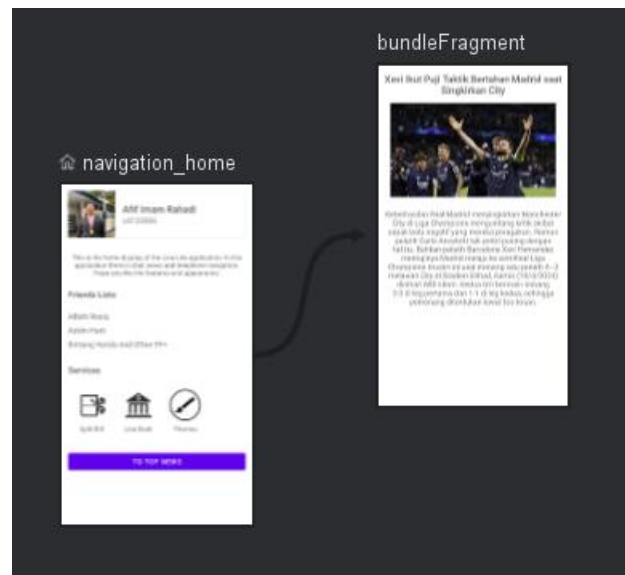
Berikut adalah hasil pada aplikasi yang telah saya buat berupa Line Lite. Aplikasi ini berupa social media untuk berkomunikasi dengan teman.

A. Home



Ini adalah halaman home dari aplikasi yang saya buat. Aplikasi saya Bernama Line Lite yang terinspirasi dari aplikasi line. Pada halaman ini di bagian atas terdapat profile saya yang berisi foto saya, nama lengkap dan juga nim. Di bawahnya terdapat teks yang mendeskripsikan aplikasi ini. Selanjutnya diikuti oleh friends list yang ada dalam akun saya. Berikutnya, adalah pelayanan apa saja yang ada di Line Lite. Akan tetapi, ini hanya visual saja dan belum ada aksi apa apa jika gambar dipencet. Dan pada bagian bawah, terdapat button menuju ke halaman top news yang mengirimkan data menggunakan bundle yang nantinya jika dipencet akan masuk ke halaman fragment bundle yang sudah saya buat.

B. Fragment Bundle (Top News)

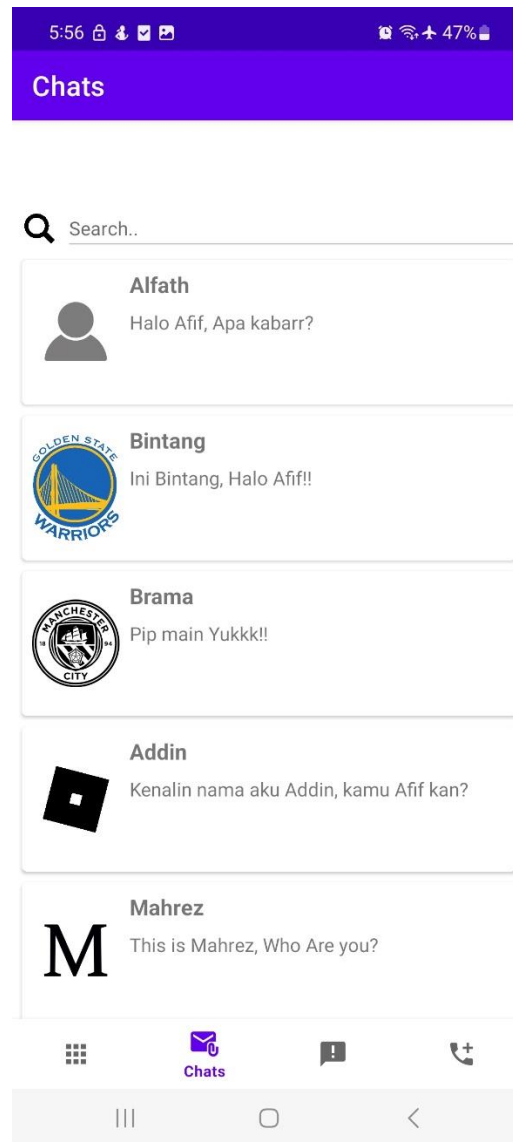


Ini adalah screenshot hasil dari navigation action yang saya lakukan. Jadi dalam halaman home terdapat pengiriman data melalui bundle yang nantinya akan masuk ke halaman bundle fragment di bawah ini.



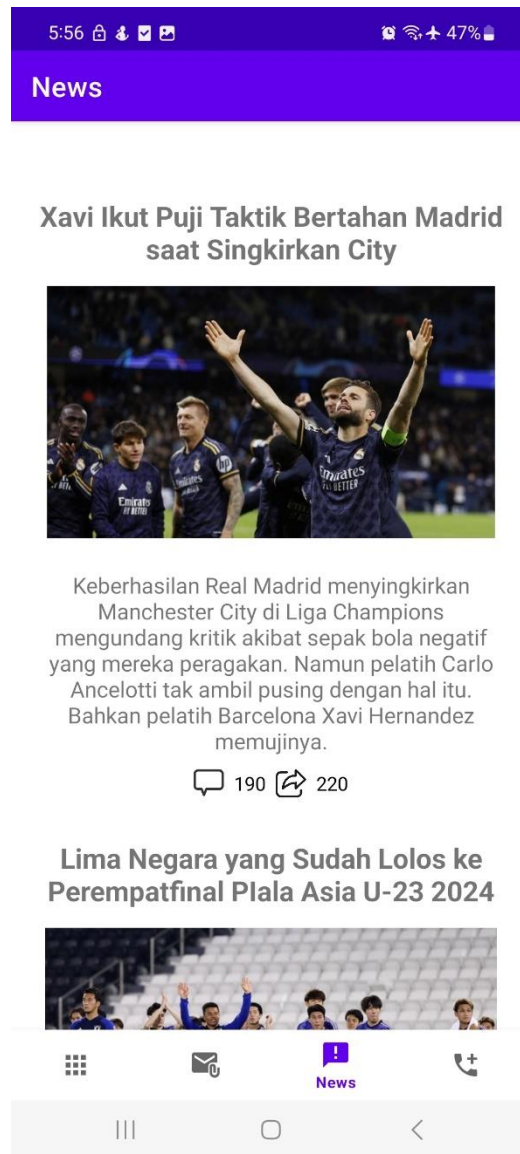
Ini adalah halaman fragment bundle yang berisi 1 berita top paling atas. Dalam halaman ini terdapat judul berita, foto berita, dan penjelasan berita secara lengkap. Kekurangannya terdapat pada tombol back, untuk kembali ke halaman home harus melalui tombol back pada button hp. Untuk button back atas belum bisa dipencet.

C. Chats



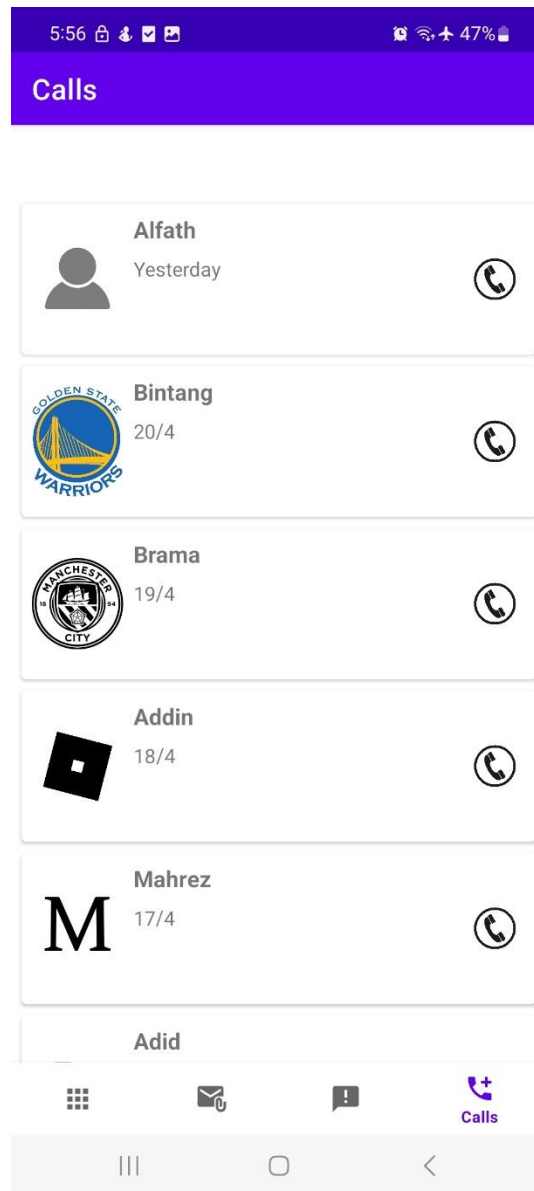
Pada halaman chats terdapat chat-chat dari teman-teman saya. Halaman ini menggunakan recycle view agar dapat melakukan scroll dan data chat terlihat banyak. Isi recycle view ini menggunakan cardview agar terlihat lebih terstruktur. Isi dari card view berupa foto profile teman, nama teman, dan isi chat terakhir. Pada setiap item chat, belum ada aksi onclick sehingga ini hanya tampilan saja. Di sini juga terdapat search box sebagai tambahan tampilan, search box ini saya disable sehingga tidak bisa dipencet atau diklik.

D. News



Ini adalah halaman news. Halaman ini terdiri dari beberapa berita yang sudah saya pilih dan terbaru. Halaman ini juga menggunakan recycle view agar pengguna dapat melakukan scroll terhadap berita. Di dalam setiap berita terdapat judul berita, foto berita, dan deskripsi berita secara singkat. Dan pada bagian bawah setiap berita terdapat icon comment dan share beserta jumlahnya. Akan tetapi, setiap comment dan share pada setiap berita berjumlah sama.

E. Calls



Tidak berbeda jauh dengan chats. Halaman calls ini juga menggunakan recycle view dan juga cardview yang sama. Akan tetapi, perbedaan terletak dari tampilan berupa pada bagian bawah nama kontak, terdapat terakhir kali teman saya menelpon saya. Dan juga pada bagian kanan terdapat icon telepon yang mengartikan bahwa ini adalah halaman call. Tidak ada aksi onclick pada halaman ini.