

LAPORAN PRAKTIKUM PENGEMBANGAN APLIKASI BERGERAK

Android Fundamental 2 - WEEK 7



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

Pada praktikum kali ini, program yang saya buat adalah list dari tim NBA dan juga tim Premier League. Setiap NBA dan Premier League masing-masing menampilkan 10 tim. Tampilan list berupa nama tim, gambar tim, dan deskripsi singkat. Program kali ini menerapkan Tab Layout dan Pager2.

A. Main Activity

MainActivity.kt

```
package com.l0122006.afifimam.week7

import ...

class MainActivity : AppCompatActivity() {

    companion object {
        @StringRes
        private val TAB_TITLES = IntArrayOf(
            R.string.tab_text_NBA,
            R.string.tab_text_PL
        )
    }

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

        val sectionsPagerAdapter = SectionsPagerAdapter(this)
        val viewPager: ViewPager2 = findViewById(R.id.view_pager)
        viewPager.adapter = sectionsPagerAdapter
        val tabs: TabLayout = findViewById(R.id.tabs)
        TabLayoutMediator(tabs, viewPager) { tab, position ->
            tab.text = resources.getString(TAB_TITLES[position])
        }.attach()
        supportActionBar?.elevation = 0f
    }
}
```

Pertama mengimpor kelas dan fungsi untuk digunakan dalam program ini. Selanjutnya, terdapat class MainActivity : AppCompatActivity(): Bagian ini adalah komponen yang menyediakan antarmuka pengguna untuk aplikasi sekaligus pengaturan logika. Berikutnya terdapat companion object: yang digunakan untuk mendefinisikan anggota statis atau konstan dalam sebuah kelas. Dalam program ini, TAB_TITLES dideklarasikan sebagai array konstan dari ID sumber daya untuk judul tab. override fun onCreate(savedInstanceState: Bundle?) { ... }: Baris ini mendeklarasikan metode bernama onCreate, yang dipanggil ketika aktivitas mulai. Ini mengambil objek Bundle sebagai parameter, yang berisi status terakhir aktivitas, jika ada. Selanjutnya, super.onCreate(savedInstanceState): untuk memastikan tugas-tugas penyiapan yang diperlukan dilakukan. setContentView(R.layout.activity_main): Baris ini mengatur tampilan konten aktivitas ke layout yang ditentukan dalam file XML activity_main.xml. val sectionsPagerAdapter = SectionsPagerAdapter(this): Baris ini membuat sebuah instance dari SectionsPagerAdapter, sebuah adapter kustom yang digunakan untuk mempopulasikan halaman-halaman dari ViewPager2. val viewPager: ViewPager2 = findViewById(R.id.view_pager): Baris ini menemukan widget ViewPager2 yang ditentukan dalam file layout XML dengan ID-nya dan menentukannya ke variabel viewPager. viewPager.adapter = sectionsPagerAdapter: Baris ini mengatur sectionsPagerAdapter sebagai

adapter untuk ViewPager2. Adapter menyediakan konten untuk ViewPager2. `val tabs: TabLayout = findViewById(R.id.tabs)`: Baris ini menemukan widget TabLayout yang ditentukan dalam file layout XML dengan ID-nya dan menentukannya ke variabel tabs. `TabLayoutMediator(tabs, viewPager) { tab, position -> ... }.attach()`: Baris ini membuat objek TabLayoutMediator, yang menghubungkan TabLayout dengan ViewPager2. Ini mengatur tab dan judul mereka berdasarkan posisi dalam ViewPager. `supportActionBar?.elevation = 0f`: Baris ini mengatur elevasi (bayangan) action bar aktivitas ke nol, menghapus bayangan drop di bawahnya.

Activity_main.xml

```
<?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"
    android:gravity="center"
    tools:context=".MainActivity">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="355dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="10dp"
        android:background="?attr/colorPrimary"
        android:contentDescription="Week 7"
        app:tabTextColor="@android:color/white" />

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/view_pager"
        android:layout_width="370dp"
        android:layout_height="match_parent"/>
</LinearLayout>
```

<LinearLayout>: adalah elemen root yang menandakan bahwa tata letak menggunakan LinearLayout sebagai wadah utama. Atribut-atributnya mengatur parameter tata letak seperti lebar, tinggi, orientasi, dan gravitasi. `tools:context=".MainActivity"`: Atribut ini memberikan konteks kelas MainActivity untuk tampilan dalam mode desain di Android Studio. <com.google.android.material.tabs.TabLayout>: Ini adalah elemen untuk menampilkan tab dalam antarmuka pengguna. Ini memiliki atribut-atribut yang mengatur tampilan dan perilaku tab. <androidx.viewpager2.widget.ViewPager2>: Ini adalah elemen yang menampilkan konten halaman yang dapat digulir secara horizontal. Ini adalah bagian dari ViewPager2, yang memungkinkan pengguna untuk menggeser antara beberapa tampilan.

B. NBA

Data class NBA.kt

```
package com.l0122006.afifimam.week7

import ...

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

Ini adalah data class dari NBA. Data class berisi name dengan tipe data string, desc dengan tipe data string, dan img dengan tipe data int. Ini digunakan untuk menyimpan data nama tim, deskripsi tim, dan gambar tim.

List NBA Adapter.kt

```
package com.l0122006.afifimam.week7

import ...

class ListNbaAdapter(private val listNba: ArrayList<Nba>) : RecyclerView.Adapter<ListNbaAdapter.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_nba, parent, attachToRoot: false)
        return ListViewHolder(view)
    }

    override fun getItemCount(): Int = listNba.size

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

Pertama, terdapat Package dan Import yang diperlukan dalam program ini. Selanjutnya, ListNbaAdapter class yang digunakan untuk menghubungkan data dari ArrayList listNba dengan RecyclerView. Ini mengimplementasikan tiga metode dari RecyclerView.Adapter: onCreateViewHolder(): Metode ini dipanggil ketika RecyclerView memerlukan sebuah ViewHolder baru. getItemCount(): Metode ini mengembalikan jumlah item dalam daftar. onBindViewHolder(): Metode ini digunakan untuk memasang data dari ArrayList ke ViewHolder di posisi tertentu. ListViewHolder class: Kelas ini adalah ViewHolder kustom yang memegang referensi ke elemen-elemen tampilan dalam satu item baris RecyclerView. Ini mengikat elemen-elemen tampilan dengan ID yang sesuai dari layout item_row_nba. onCreateViewHolder() method: Metode ini membuat dan mengembalikan instance dari ListViewHolder. Itu mengisi tata letak dari file XML item_row_nba ke dalam tampilan. getItemCount() method: Metode ini mengembalikan jumlah item dalam daftar listNba.

onBindViewHolder() method: Metode ini menghubungkan data dari objek Nba dalam listNba ke elemen-elemen tampilan dalam ViewHolder pada posisi tertentu. Ini menetapkan gambar, nama, dan deskripsi dari objek Nba ke ImageView dan TextView yang sesuai dalam ViewHolder.

NBA Fragment.kt

```
package com.l0122006.afifimam.week7

import ...

class NBAFragment : Fragment() {
    private lateinit var rvNbas: RecyclerView
    private val list = ArrayList<Nba>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val rootView = inflater.inflate(R.layout.fragment_n_b_a, container, attachToRoot: false)

        rvNbas = rootView.findViewById(R.id.rv_nba)
        rvNbas.setHasFixedSize(true)

        list.addAll(getListNbas())
        showRecyclerList()

        return rootView
    }

    private fun getListNbas(): ArrayList<Nba> {
        val dataName = resources.getStringArray(R.array.data_name_nba)
        val dataDesc = resources.getStringArray(R.array.data_desc_nba)
        val dataImg = resources.obtainTypedArray(R.array.data_img_nba)
        val listNba = ArrayList<Nba>()
        for (i in dataName.indices) {
            val nba = Nba(dataName[i], dataDesc[i], dataImg.getResourceId(i, defValue: -1))
            listNba.add(nba)
        }
        return listNba
    }
}
```

NBAFragment class: Kelas ini adalah subclass dari Fragment, yang merepresentasikan satu bagian modular dari antarmuka pengguna atau perilaku dalam sebuah Activity. Ini mengimplementasikan metode-metode yang diperlukan untuk mengelola fragment.onCreateView() method: Metode ini dipanggil saat fragment harus menyediakan tampilan untuk ditampilkan. Ini menginflasi layout dari file XML fragment_n_b_a yang didefinisikan untuk fragment ini. Kemudian, RecyclerView diinisialisasi dan diatur dengan adapter untuk menampilkan daftar item. getListNbas() method: Metode ini digunakan untuk mendapatkan daftar objek Nba dari sumber daya string dan array gambar di file arrays.xml. Ini membuat objek Nba dari array nama, deskripsi, dan ID gambar, dan menambahkannya ke dalam ArrayList. showRecyclerList() method: Metode ini mengatur RecyclerView dengan LayoutManager LinearLayoutManager dan mengatur adapter ListNbaAdapter untuk menampilkan daftar item. companion object: Bagian ini mendefinisikan metode newInstance() yang merupakan cara konvensional untuk membuat instance baru dari fragment.

Item_row_nba.xml

```
<?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"
    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:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        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"
            tools:src="@tools:sample/avatars" />
```

<androidx.cardview.widget.CardView>: Ini adalah elemen root yang menandakan bahwa tampilan ini akan dikelilingi oleh CardView. CardView adalah tampilan yang menampilkan konten dalam kartu dengan bayangan. android:id="@+id/card_view": Ini adalah ID unik untuk CardView. Selanjutnya, ada juga pengatur tinggi dan juga lebarnya sesuai dengan keinginan sendiri. android:layout_marginStart, android:layout_marginTop, android:layout_marginEnd, dan android:layout_marginBottom: Ini adalah margin untuk CardView, mengatur jarak antara CardView dan elemen-elemen lainnya. card_view:cardCornerRadius="4dp": Ini mengatur radius sudut CardView. <androidx.constraintlayout.widget.ConstraintLayout>: Ini adalah wadah untuk mengatur tata letak elemen-elemen dalam CardView menggunakan ConstraintLayout. <ImageView>: Ini adalah elemen untuk menampilkan gambar dalam CardView. Itu memiliki atribut-atribut yang mengatur ukuran, jenis skala, dan posisi gambar. <TextView>: Ini adalah elemen untuk menampilkan teks dalam CardView. Dua TextView digunakan di sini untuk menampilkan nama dan deskripsi item. Mereka memiliki atribut-atribut yang mengatur tata letak dan penampilan teks.

Fragment_n_b_a.xml

```
<?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=".NBAFragment">

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

</FrameLayout>
```

<FrameLayout>: Ini adalah elemen root yang menandakan bahwa tampilan fragment ini akan dikelilingi oleh FrameLayout. `android:layout_width="match_parent"` dan `android:layout_height="match_parent"`: Ini mengatur lebar dan tinggi FrameLayout agar mengisi lebar dan tinggi parent-nya, sehingga tampilan fragment akan menempati seluruh ruang layar. `tools:context=".NBAFragment"`: Atribut ini memberikan konteks kelas NBAFragment untuk tampilan dalam mode desain di Android Studio. <androidx.recyclerview.widget.RecyclerView>: Ini adalah elemen yang digunakan untuk menampilkan daftar item dalam tampilan daftar gulir. Ini adalah bagian dari AndroidX RecyclerView library. RecyclerView diidentifikasi dengan ID `rv_nba` untuk referensi dalam kode Kotlin.

C. Premier League

Data class Premier League

```
package com.l0122006.afifimam.week7

import ...

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

Sama seperti NBA, premier league memiliki data class untuk datanya sendiri. Data class terdiri dari name, desc, dan img untuk menampilkan datanya.

ListPremierLeagueAdapter.kt

```
package com.l0122006.afifiman.week7

import ...

class ListPremierLeagueAdapter(private val listPremierLeague: ArrayList<PremierLeague>) : RecyclerView.Adapter<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 = LayoutInflater.from(parent.context).inflate(R.layout.item_row_premier_league, parent, attachToRoot = false)
        return ListViewHolder(view)
    }

    override fun getItemCount(): Int = listPremierLeague.size

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

Sama halnya dengan adapter pada NBA, pertama deklarasi Package dan Import diperlukan untuk menggunakan kelas-kelas Android dan RecyclerView. Selanjutnya, terdapat ListPremierLeagueAdapter class: Kelas ini adalah adapter kustom yang menghubungkan data dari ArrayList listPremierLeague dengan RecyclerView. Ini mengimplementasikan tiga metode dari RecyclerView.Adapter: onCreateViewHolder(): Metode ini dipanggil ketika RecyclerView memerlukan sebuah ViewHolder baru. getItemCount(): Metode ini mengembalikan jumlah item dalam daftar. onBindViewHolder(): Metode ini digunakan untuk memasang data dari ArrayList ke ViewHolder di posisi tertentu. ListViewHolder class: Kelas ini adalah ViewHolder kustom yang memegang referensi ke elemen-elemen tampilan dalam satu item baris RecyclerView. Ini mengikat elemen-elemen tampilan dengan ID yang sesuai dari layout item_row_premier_league. onCreateViewHolder() method: Metode ini membuat dan mengembalikan instance dari ListViewHolder. Itu mengisi tata letak dari file XML item_row_premier_league ke dalam tampilan. getItemCount() method: Metode ini mengembalikan jumlah item dalam daftar listPremierLeague. onBindViewHolder() method: Metode ini menghubungkan data dari objek PremierLeague dalam listPremierLeague ke elemen-elemen tampilan dalam ViewHolder pada posisi tertentu. Ini menetapkan gambar, nama, dan deskripsi dari objek PremierLeague ke ImageView dan TextView yang sesuai dalam ViewHolder.

PremierLeagueFragment

```
package com.l0122006.afifimam.week7

import ...

class PremierLeagueFragment : Fragment() {
    private lateinit var rvPremierLeagues: RecyclerView
    private val list = ArrayList<PremierLeague>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val rootView = inflater.inflate(R.layout.fragment_premier_league, container, attachToRoot false)

        rvPremierLeagues = rootView.findViewById(R.id.rv_premier_league)
        rvPremierLeagues.setHasFixedSize(true)

        list.addAll(getListPremierLeagues())
        showRecyclerList()

        return rootView
    }

    private fun getListPremierLeagues(): ArrayList<PremierLeague> {
        val dataName = resources.getStringArray(R.array.data_name_premier_league)
        val dataDesc = resources.getStringArray(R.array.data_desc_premier_league)
        val dataImg = resources.obtainTypedArray(R.array.data_img_premier_league)
        val listPremierLeague = ArrayList<PremierLeague>()
        for (i in dataName.indices) {
            val premierLeague = PremierLeague(dataName[i], dataDesc[i], dataImg.getResourceId(i, defValue: -1))
            listPremierLeague.add(premierLeague)
        }
        return listPremierLeague
    }
}
```

Package dan Import: Deklarasi package dan impor yang diperlukan untuk menggunakan kelas-kelas Android dan RecyclerView. PremierLeagueFragment class: Kelas ini adalah subclass dari Fragment, yang merepresentasikan satu bagian modular dari antarmuka pengguna atau perilaku dalam sebuah Activity. Ini mengimplementasikan metode-metode yang diperlukan untuk mengelola fragment. onCreateView() method: Metode ini dipanggil saat fragment harus menyediakan tampilan untuk ditampilkan. Ini menginflasi layout dari file XML fragment_premier_league yang didefinisikan untuk fragment ini. Kemudian, RecyclerView diinisialisasi dan diatur dengan adapter untuk menampilkan daftar item. getListPremierLeagues() method: Metode ini digunakan untuk mendapatkan daftar objek PremierLeague dari sumber daya string dan array gambar di file arrays.xml. Ini membuat objek PremierLeague dari array nama, deskripsi, dan ID gambar, dan menambahkannya ke dalam ArrayList. showRecyclerList() method: Metode ini mengatur RecyclerView dengan LayoutManager LinearLayoutManager dan mengatur adapter ListPremierLeagueAdapter untuk menampilkan daftar item. companion object: Bagian ini mendefinisikan metode newInstance() yang merupakan cara konvensional untuk membuat instance baru dari fragment.

Item_row_premier_league.xml

```
<?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/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:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        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"
            tools:src="@tools:sample/avatars" />
```

<androidx.cardview.widget.CardView>: Ini adalah elemen root yang menandakan bahwa tampilan ini akan dikelilingi oleh CardView. CardView adalah tampilan yang menampilkan konten dalam kartu dengan bayangan. android:id="@+id/card_view": Ini adalah ID unik untuk CardView. Terdapat pengaturan lebar dan tinggi sesuai dengan keinginan user agar tampilan lebih menarik. Ini mengatur lebar CardView agar mengisi lebar parent dan tinggi agar menyesuaikan dengan konten di dalamnya. android:layout_gravity="center": Ini mengatur gravitasi CardView menjadi di tengah. android:layout_marginStart, android:layout_marginTop, android:layout_marginEnd, dan android:layout_marginBottom: Ini adalah margin untuk CardView, mengatur jarak antara CardView dan elemen-elemen lainnya. card_view:cardCornerRadius="4dp": Ini mengatur radius sudut CardView.

<androidx.constraintlayout.widget.ConstraintLayout>: Ini adalah wadah untuk mengatur tata letak elemen-elemen dalam CardView menggunakan ConstraintLayout.

<ImageView>: Ini adalah elemen untuk menampilkan gambar dalam CardView. Itu memiliki atribut-atribut yang mengatur ukuran, jenis skala, dan posisi gambar.

<TextView>: Ini adalah elemen untuk menampilkan teks dalam CardView. Dua TextView digunakan di sini untuk menampilkan nama dan deskripsi item. Mereka memiliki atribut-atribut yang mengatur tata letak dan penampikan teks.

Fragment_premier_league

```
<?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=".PremierLeagueFragment">

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

</FrameLayout>
```

<FrameLayout>: Ini adalah elemen root yang menandakan bahwa tampilan fragment ini akan dikelilingi oleh FrameLayout. FrameLayout adalah tata letak yang sederhana, yang memungkinkan satu anak tampilan ditempatkan di atas yang lainnya. `android:layout_width="match_parent"` dan `android:layout_height="match_parent"`: Ini mengatur lebar dan tinggi FrameLayout agar mengisi lebar dan tinggi parent-nya, sehingga tampilan fragment akan menempati seluruh ruang layar. `tools:context=".PremierLeagueFragment"`: Atribut ini memberikan konteks kelas PremierLeagueFragment untuk tampilan dalam mode desain di Android Studio. <androidx.recyclerview.widget.RecyclerView>: Ini adalah elemen yang digunakan untuk menampilkan daftar item dalam tampilan daftar gulir. Ini adalah bagian dari AndroidX RecyclerView library. RecyclerView diidentifikasi dengan ID `rv_premier_league` untuk referensi dalam kode Kotlin.

D. Section Pager Adapter

```
package com.t0122006.afifimam.week7

import ...

class SectionPagerAdapter(activity: AppCompatActivity) : FragmentStateAdapter(activity) {
    override fun getItemCount(): Int {
        return 2
    }

    override fun createFragment(position: Int): Fragment {
        var fragment: Fragment? = null
        when (position) {
            0 -> fragment = NBAFragment()
            1 -> fragment = PremierLeagueFragment()
        }
        return fragment as Fragment
    }
}
```

Kode di atas adalah implementasi dari kelas SectionPagerAdapter yang merupakan subclass dari FragmentStateAdapter dalam aplikasi Android. Ini bertanggung jawab untuk mengatur dan menyediakan fragmen untuk ditampilkan dalam ViewPager2.

2. Screenshot Terminal

Berikut adalah hasil pada emulator terkait program yang telah saya buat

A. NBA Team



Ini adalah hasil dari page NBA Teams. Tampilan berupa nama tim, gambar tim, dan deskripsi singkat. Tampilan menggunakan recycle view agar dapat melakukan scroll untuk setiap datanya.

B. Premier League Team



Ini adalah hasil dari page Premier League Teams. Tampilan berupa nama tim, gambar tim, dan deskripsi singkat. Tampilan menggunakan recycle view agar dapat melakukan scroll untuk setiap datanya.