

**UJIAN AKHIR SEMESTER
BAHASA PEMOGRAMAN II
S1-SISTEM INFORMASI**

“Aplikasi Pengelolaan Barang E-Commerce”



Disusun oleh :

Dinda Farras Ghinannafsi / 21.12.2170 (Ketua)

Oktaviana Arifah / 21.12.2147

Dzaka Zhilal / 21.12.2146

Ibnu Arbianto / 21.12.2143

Desy Ikasari / 21.12.2141

**S1 – SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
2022/2023**

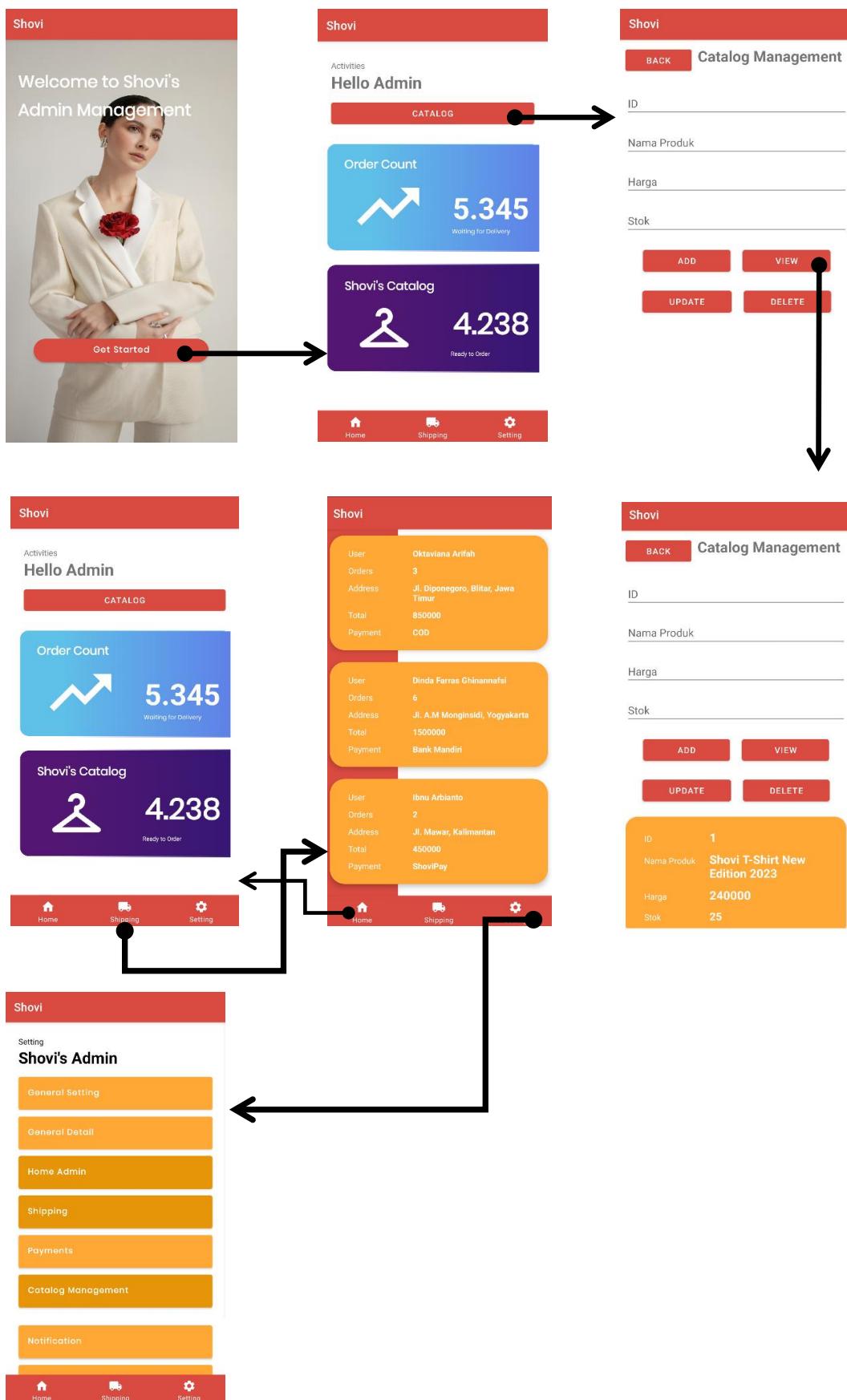
BAB I

ABSTRAKSI

Pada Final Project kali ini kami membuat sebuah aplikasi pengelolaan barang dari brand yang kami berinama Shovi. Shovi merupakan elektronik komersial yang mengangkat nama brandnya sendiri, dengan kata lain aplikasi ini hanya menjual product dari brand Shovi saja. Pada kesempatan kali ini Point of View pada aplikasi yang kami buat yaitu sebagai Admin dari Brand Shovi, yang mana seorang admin dapat mengelola produk yang dijual, baik menambah barang, mengupdate barang, menghapus barang, atau mungkin hanya melihat barang yang sedang tersedia di catalog.

Tujuan dari kami membuat aplikasi ini yaitu untuk memudahkan Admin dalam mengelola produk yang ada. Desain dari aplikasi ini juga kami rancang se sederhana mungkin, sehingga tidak menyulitkan Admin.

Berikut kami berikan alur dari aplikasi yang kami buat.



BAB II

PENJENJELASAN LAYOUT

A. activity_main.xml

Layout ini merupakan layout atau design dari tampilan pertama kali membuka aplikasi. Pada tampilan ini terdapat satu button yang apabila di klik, button tersebut akan menuju ke halaman home admin.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:focusedInTouchMode="true"
8      tools:context=".MainActivity"
9      tools:ignore="SpeakableTextPresentCheck">
10
11     <ImageView
12         android:id="@+id/bg"
13         android:layout_width="500dp"
14         android:layout_height="match_parent"
15         android:layout_alignParentLeft="true"
16         android:layout_alignParentTop="true"
17         android:background="@drawable/login"
18         android:contentDescription="T000"
19         app:layout_constraintBottom_toBottomOf="parent"
20         app:layout_constraintEnd_toEndOf="parent"
21         app:layout_constraintHorizontal_bias="0.505"
22         app:layout_constraintStart_toStartOf="parent"
23         app:layout_constraintTop_toTopOf="parent"
24         app:layout_constraintVertical_bias="1.0"
25         tools:ignore="MissingConstraints" />
26
27     <TextView
28         android:id="@+id/textView"
29         android:layout_width="300dp"
30
31         tools:ignore="MissingConstraints" />
32
33     <Textview
34         android:id="@+id/textView"
35         android:layout_width="300dp"
36         android:layout_height="wrap_content"
37         android:layout_marginStart="24dp"
38         android:layout_marginLeft="24dp"
39         android:layout_marginTop="50dp"
40         android:layout_marginEnd="100dp"
41         android:layout_marginRight="100dp"
42         android:fontFamily="@font/poppins"
43         android:lineSpacingExtra="8sp"
44         android:text="Welcome to Shovi's Admin Management"
45         android:textColor="#fff"
46         android:textSize="28sp"
47         android:textStyle="bold"
48
49         app:layout_constraintBottom_toTopOf="@+id/linearLayout"
50         app:layout_constraintEnd_toEndOf="parent"
51         app:layout_constraintHorizontal_bias="0.0"
52         app:layout_constraintStart_toStartOf="parent"
53         app:layout_constraintTop_toTopOf="parent"
54         app:layout_constraintVertical_bias="0.0"
55         tools:ignore="MissingConstraints" />
56
57     <LinearLayout
58         android:id="@+id/linearLayout"
59         android:layout_width="match_parent"
60
61         android:id="@+id/linearLayout"
62         android:layout_width="match_parent"
63         android:layout_height="51dp"
64         android:layout_marginTop="32dp"
65         android:layout_marginBottom="134dp"
66         android:orientation="horizontal"
67
68         app:layout_constraintBottom_toBottomOf="@+id/bg"
69         app:layout_constraintEnd_toEndOf="parent"
70         app:layout_constraintStart_toStartOf="parent"
71         >
72
73     <Button
74         android:id="@+id/btn_login"
75         android:layout_width="120dp"
76         android:layout_height="match_parent"
77         android:layout_marginLeft="56dp"
78         android:layout_marginRight="56dp"
79         android:layout_weight="1"
80         android:background="@#094440"
81         android:fontFamily="@font/poppins"
82         android:text="Get Started"
83         android:textAllCaps="false"
84         android:textStyle="bold"
85         app:cornerRadius="20dp" />
86
87     </LinearLayout>
88 </androidx.constraintlayout.widget.ConstraintLayout>
```

Kami menggunakan layout constraint sebagai default dari halaman ini.

- Baris 11-25

Kami menggunakan tag ImageView untuk menampilkan gambar pada background. Kami memberikan id dengan nama ‘bg’. Cara kami memasang gambar yang kami inginkan yaitu dengan memanggil package drawable diikuti nama gambarnya yang sudah kami simpan (android:background=”@drawable/login”)

- Baris 27-49

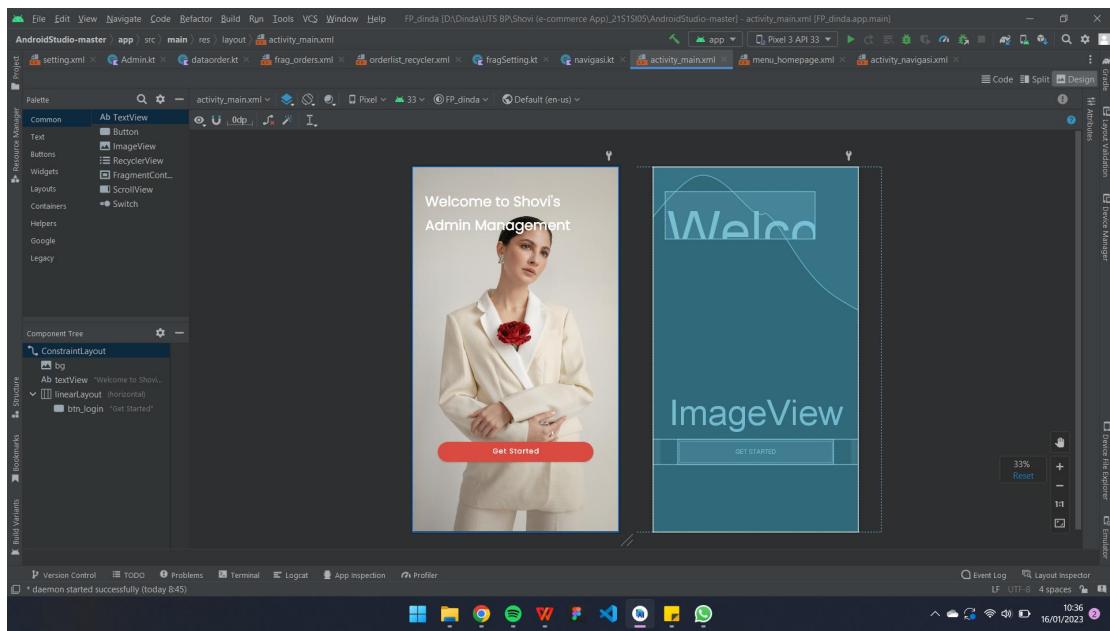
Menggunakan tag TextView hanya sebagai hiasan, kami menggunakan font poppins, dan memberikan id dengan nama ‘textView’

- Baris 51-61

Menggunakan tag LinearLayout untuk membuat layout secara linear yang nantinya pada layout tersebut berisi tag button

- Baris 64-76

Menggunakan tag Button untuk membuat button. Kami memberi id button tersebut dengan nama btn_login. Kami menggunakan font poppins, lalu pada button tersebut terdapat tulisan Get Started



B. activity_navigasi.xml

Layout ini merupakan sebagai design dari navigasi yang sudah dibuat di package menu, file menu_homepage.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".navigasi">
8
9     <com.google.android.material.bottomnavigation.BottomNavigationView
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:id="@+id/bottom_navigation"
13         android:background="#D9A480"
14         app:menu="@menu/menu_homepage"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintLeft_toLeftOf="parent"
17         app:layout_constraintRight_toRightOf="parent"
18         app:itemIconTint="#fffff"
19         app:itemTextColor="#fffff"/>
20
21     <FrameLayout
22         android:id="@+id/fragment_container"
23         android:layout_width="match_parent"
24         android:layout_height="0dp"
25         app:layout_constraintTop_toTopOf="parent"
26         app:layout_constraintBottom_toTopOf="@+id/bottom_navigation"
27     />
28 </androidx.constraintlayout.widget.ConstraintLayout>

```

Kami menggunakan layout constraint sebagai default dari halaman ini. Layout ini menyambung dengan file navigasi.kt.

- Baris 9-19

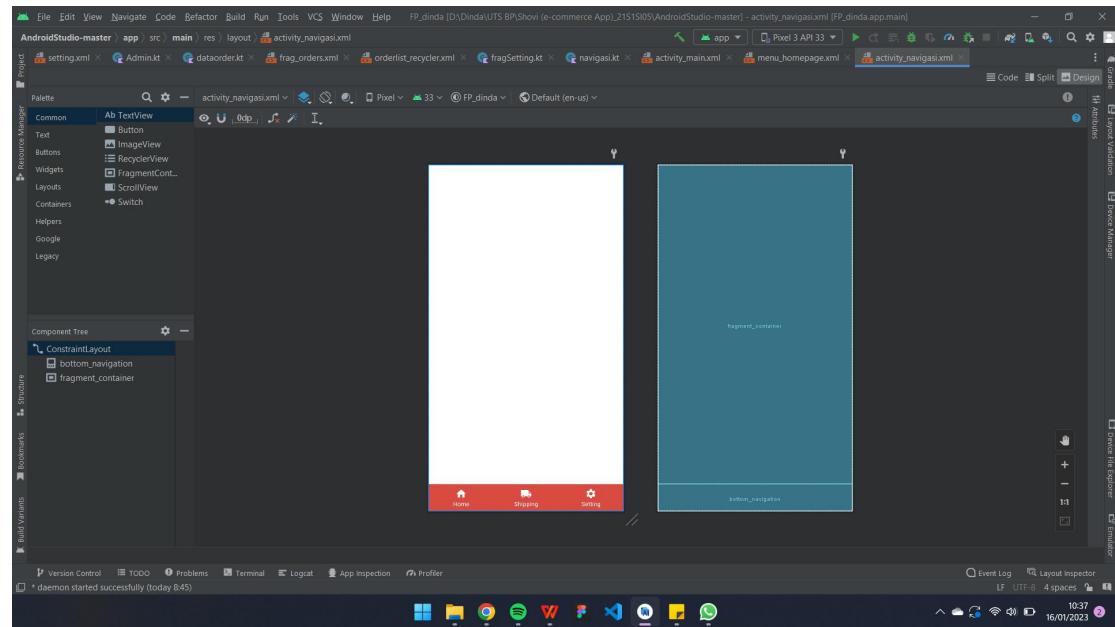
Menggunakan **BottomNavigationView** tag
com.google.android.material.bottomnavigation.BottonNavigationView agar letaknya

berada dibawah layaknya navigasi pada aplikasi. Menyambungkan layout ini dengan layout menu_homepage di package menu (app:menu="@menu/menu_homepage")

- Baris 21-27

Menggunakan FrameLayout untuk menampilkan fragment homeadmin. Memberi id dengan nama fragment_container

UI



C. frag_home_admin.xml

Layout ini merupakan sebagai design dari halaman home admin yang merupakan fragment.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".fragment.fraghomeAdmin">
8
9      <LinearLayout
10         android:id="@+id/linearLayout3"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:layout_marginStart="24dp"
14         android:layout_marginLeft="24dp"
15         android:layout_marginTop="31dp"
16         android:layout_marginEnd="24dp"
17         android:layout_marginRight="24dp"
18         android:layout_marginBottom="658dp"
19         android:orientation="vertical"
20         app:layout_constraintBottom_toBottomOf="@+id/constraintLayout"
21         app:layout_constraintEnd_toEndOf="parent"
22         app:layout_constraintStart_toStartOf="parent"
23         app:layout_constraintTop_toTopOf="parent">
24
25             <TextView
26                 android:layout_width="match_parent"
27                 android:layout_height="wrap_content"
28                 android:text="Activities"
29                 android:textSize="14sp" />

```

```

31   <TextView
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:layout_marginBottom="10dp"
35     android:text="Hello Admin"
36     android:textSize="28dp"
37     android:textStyle="bold" />
38
39   <Button
40     android:id="@+id/tes1"
41     android:layout_width="match_parent"
42     android:layout_height="wrap_content"
43     android:text="Catalog"
44     app:backgroundTint="#D9A440" />
45
46   <ImageView
47     android:id="@+id/shipping"
48     android:layout_width="wrap_content"
49     android:layout_height="wrap_content"
50     android:layout_marginTop="28dp"
51     android:src="@drawable/kotak"
52     app:layout_constraintEnd_toEndOf="parent"
53     app:layout_constraintHorizontal_bias="0.529"
54     app:layout_constraintStart_toStartOf="parent"
55     app:layout_constraintTop_toBottomOf="@+id/linearLayout" />
56
57

```

```

58   <ImageView
59     android:id="@+id/shingrad"
60     android:layout_width="290dp"
61     android:layout_height="184dp"
62     android:src="@drawable/gradias"
63     app:layout_constraintBottom_toBottomOf="@+id/shipping"
64     app:layout_constraintEnd_toEndOf="@+id/shipping"
65     app:layout_constraintHorizontal_bias="1.0"
66     app:layout_constraintStart_toStartOf="@+id/shipping"
67     app:layout_constraintTop_toTopOf="@+id/shipping"
68     app:layout_constraintVertical_bias="0.0" />
69
70
71   <TextView
72     android:id="@+id/txt_shipping"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:fontFamily="@font/poppins"
76     android:text="Order Count"
77     android:textColor="@color/white"
78     android:textSize="20dp"
79     android:textStyle="bold"
80     app:layout_constraintBottom_toBottomOf="@+id/shipping"
81     app:layout_constraintEnd_toEndOf="@+id/shipping"
82     app:layout_constraintHorizontal_bias="0.122"
83     app:layout_constraintStart_toStartOf="@+id/shipping"
84     app:layout_constraintTop_toTopOf="@+id/shipping"
85     app:layout_constraintVertical_bias="0.136" />
86
87

```

● Baris 29-43

Menggunakan tag Button untuk membuat button, pada button tersebut dituliskan Catalog, yang mana apabila button tersebut diklik nantinya akan menuju halaman pengelolaan barang. Button ini kami beri id dengan nama tes1

```

87   <ImageView
88     android:id="@+id/shippinglog"
89     android:layout_width="127dp"
90     android:layout_height="148dp"
91     android:layout_marginStart="40dp"
92     android:layout_marginLeft="40dp"
93     android:layout_marginEnd="200dp"
94     android:layout_marginRight="200dp"
95     android:src="@drawable/ic_baseline_trending_up_24"
96     app:layout_constraintBottom_toBottomOf="@+id/imageView2"
97     app:layout_constraintEnd_toEndOf="@+id/imageView2"
98     app:layout_constraintStart_toStartOf="@+id/shipping"
99     app:layout_constraintTop_toTopOf="@+id/shipping"
100    app:layout_constraintVertical_bias="1.0" />
101
102
103  <TextView
104    android:id="@+id/countship"
105    android:layout_width="wrap_content"
106    android:layout_height="wrap_content"
107    android:layout_marginStart="213dp"
108    android:layout_marginLeft="213dp"
109    android:layout_marginTop="74dp"
110    android:layout_marginEnd="18dp"
111    android:layout_marginRight="18dp"
112    android:layout_marginBottom="43dp"
113    android:textColor="5, 345"
114    android:textSize="50dp"
115    android:textStyle="bold" />

```

```

128
129   <TextView
130     android:id="@+id/shipdes"
131     android:layout_width="wrap_content"
132     android:layout_height="wrap_content"
133     android:layout_marginStart="143dp"
134     android:layout_marginLeft="143dp"
135     android:layout_marginTop="141dp"
136     android:layout_marginEnd="51dp"
137     android:layout_marginRight="51dp"
138     android:layout_marginBottom="23dp"
139     android:fontFamily="@font/poppins"
140     android:text="Waiting for Delivery"
141     android:textColor="@color/white"
142     android:textSize="10dp"
143     app:layout_constraintBottom_toBottomOf="@+id/shipping"
144     app:layout_constraintEnd_toEndOf="@+id/shingrad"
145     app:layout_constraintStart_toStartOf="@+id/shipgrad"
146     app:layout_constraintTop_toTopOf="@+id/shipping" />
147
148   <ImageView
149     android:id="@+id/imageView3"
150     android:layout_width="wrap_content"
151     android:layout_height="wrap_content"
152     android:layout_marginTop="25dp"
153     android:src="@drawable/kotak1"
154     app:layout_constraintBottom_toBottomOf="parent"
155     app:layout_constraintEnd_toEndOf="parent"
156     app:layout_constraintHorizontal_bias="0.529" />

```

```

153   <ImageView
154     android:id="@+id/imageView4"
155     android:layout_width="294dp"
156     android:layout_height="289dp"
157     android:src="@drawable/gradient2"
158     app:layout_constraintBottom_toBottomOf="@+id/imageView3"
159     app:layout_constraintEnd_toEndOf="@+id/imageView3"
160     app:layout_constraintHorizontal_bias="1.0"
161     app:layout_constraintStart_toStartOf="@+id/imageView3"
162     app:layout_constraintTop_toTopOf="@+id/imageView3" />
163
164   <TextView
165     android:id="@+id/txt_catalog"
166     android:layout_width="wrap_content"
167     android:layout_height="wrap_content"
168     android:fontFamily="@font/poppins"
169     android:text="Shovi's Catalog"
170     android:textColor="@color/white"
171     android:textSize="20dp"
172     android:textStyle="bold"
173     app:layout_constraintBottom_toBottomOf="@+id/imageView3"
174     app:layout_constraintEnd_toEndOf="@+id/imageView3"
175     app:layout_constraintHorizontal_bias="0.141"
176     app:layout_constraintStart_toStartOf="@+id/imageView3"
177     app:layout_constraintTop_toTopOf="@+id/imageView3"
178     app:layout_constraintVertical_bias="0.143" />
179

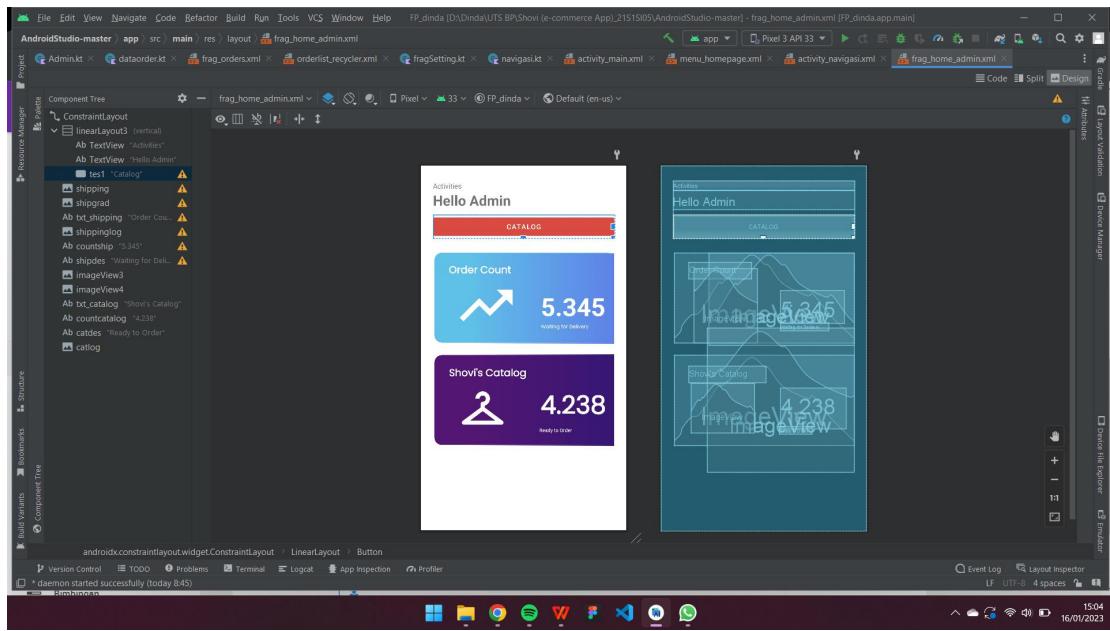
```

```

180     <TextView
181         android:id="@+id/countcatalog"
182         android:layout_width="wrap_content"
183         android:layout_height="wrap_content"
184         android:layout_marginStart="213dp"
185         android:layout_marginLeft="213dp"
186         android:layout_marginTop="7dp"
187         android:layout_marginEnd="18dp"
188         android:layout_marginRight="18dp"
189         android:layout_marginBottom="43dp"
190         android:fontFamily="@font/poppins"
191         android:text="4.238"
192         android:textColor="@color/white"
193         android:textSize="50dp"
194         android:textStyle="bold"
195         app:layout_constraintBottom_toBottomOf="@+id/imageView3"
196         app:layout_constraintEnd_toEndOf="@+id/imageView4"
197         app:layout_constraintStart_toStartOf="@+id/imageView3"
198         app:layout_constraintTop_toTopOf="@+id/imageView3" />
199
200     <TextView
201         android:id="@+id/catdes"
202         android:layout_width="wrap_content"
203         android:layout_height="wrap_content"
204         android:layout_marginStart="143dp"
205         android:layout_marginLeft="143dp"
206         android:layout_marginTop="141dp"
207         android:layout_marginEnd="51dp"
208         android:layout_marginRight="51dp"
209
210         android:layout_marginRight="51dp"
211         android:layout_marginBottom="23dp"
212         android:text="Ready to Order"
213         android:textColor="@color/white"
214         android:textSize="10dp"
215         app:layout_constraintBottom_toBottomOf="@+id/imageView3"
216         app:layout_constraintEnd_toEndOf="@+id/catlog"
217         app:layout_constraintStart_toStartOf="@+id/catlog"
218         app:layout_constraintTop_toTopOf="@+id/imageView3"
219         app:layout_constraintVertical_bias="1.0" />
220
221     <ImageView
222         android:id="@+id/catlog"
223         android:layout_width="127dp"
224         android:layout_height="108dp"
225         android:layout_marginStart="40dp"
226         android:layout_marginLeft="40dp"
227         android:layout_marginEnd="200dp"
228         android:layout_marginRight="200dp"
229         android:src="@drawable/ic_baseline_checkroom_24"
230         app:layout_constraintBottom_toBottomOf="@+id/imageView3"
231         app:layout_constraintEnd_toEndOf="@+id/imageView3"
232         app:layout_constraintHorizontal_bias="1.0"
233         app:layout_constraintStart_toStartOf="@+id/imageView3"
234         app:layout_constraintTop_toTopOf="@+id/imageView3"
235         app:layout_constraintVertical_bias="0.7" />
236
237     </androidx.constraintlayout.widget.ConstraintLayout>

```

UI



D. frag_orders.xml

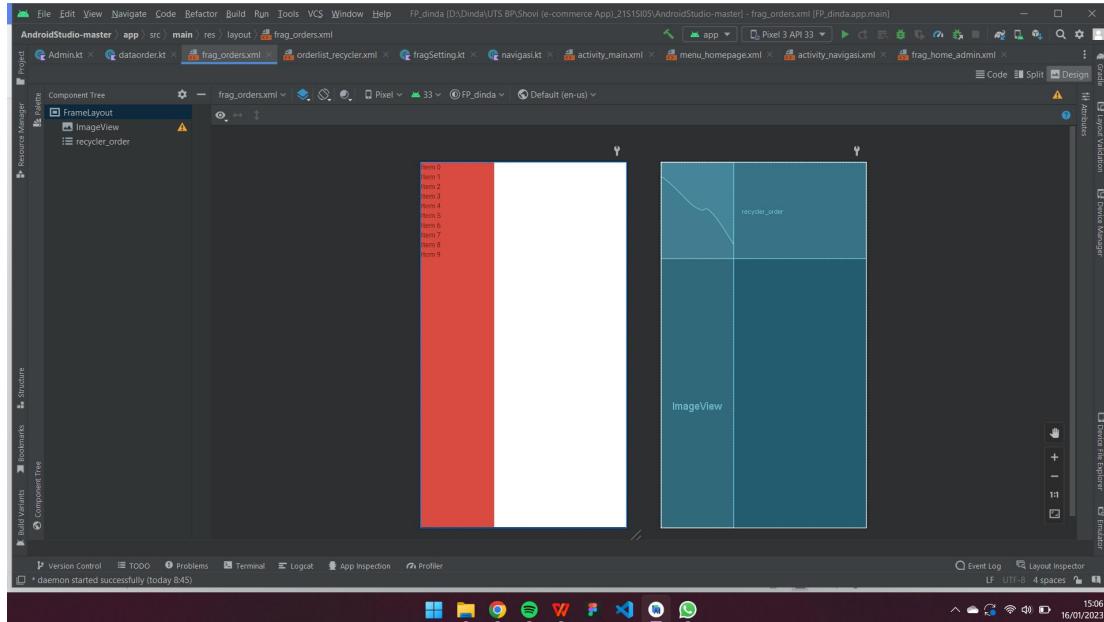
Layout ini merupakan design dari fragment order, dimana recycler dari data yang sudah kita buat akan ditampilkan di halaman ini. Halaman ini menyambung dengan file fragorders.kt

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".fragment.fragorders">
8     <ImageView
9         android:layout_width="match_parent"
10        android:layout_height="match_parent"
11        android:src="@drawable/kotak2"
12        android:layout_marginRight="265dp"/>
13     <androidx.recyclerview.widget.RecyclerView
14         android:id="@+id/recycler_order"
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content">
17     </androidx.recyclerview.widget.RecyclerView>
18 </FrameLayout>
19

```

UI



Menggunakan tag FrameLayout agar beberapa komponen dapat saling tumpang tindih.

- Baris 8-12

Menggunakan tag ImageView untuk menampilkan gambar persegi panjang merah sebagai hiasan

- Baris 13-17

Menggunakan tag recycler untuk menampilkan list recycler yang sudah dibuat

E. orderlist_recycler.xml

Layout ini merupakan tampilan dari recycler yang akan muncul di halaman frag_orders.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="wrap_content"
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      app:cardBackgroundColor="#ffa836"
7      app:cardCornerRadius="25dp"
8      app:cardElevation="5dp"
9      android:layout_margin="10dp">
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:orientation="vertical"
14         android:padding="16dp">
15             <LinearLayout
16                 android:layout_width="match_parent"
17                 android:layout_height="wrap_content"
18                 android:orientation="horizontal"
19                 android:layout_margin="5dp">
20
21                 <androidx.appcompat.widget.AppCompatTextView
22                     android:layout_width="match_parent"
23                     android:layout_height="wrap_content"
24                     android:layout_weight="2"
25                     android:layout_marginLeft="10dp"
26                     android:text="User"
27                     android:textColor="@color/white"/>
28                 <androidx.appcompat.widget.AppCompatTextView
29                     android:id="@+id/or_name"

```

```

37
38         </LinearLayout>
39         <LinearLayout
40             android:layout_width="match_parent"
41             android:layout_height="wrap_content"
42             android:layout_margin="5dp"
43             android:orientation="horizontal">
44             <androidx.appcompat.widget.AppCompatTextView
45                 android:layout_width="match_parent"
46                 android:layout_height="wrap_content"
47                 android:layout_weight="2"
48                 android:text="Orders"
49                 android:layout_marginLeft="10dp"
50                 android:textColor="@color/white"/>
51             <androidx.appcompat.widget.AppCompatTextView
52                 android:id="@+id/or_item"
53                 android:layout_width="match_parent"
54                 android:layout_height="wrap_content"
55                 android:layout_weight="1"
56                 android:layout_marginLeft="10dp"
57                 android:text="Items"
58                 android:textColor="@color/white"
59                 android:textStyle="bold"/>
60         </LinearLayout>
61         <LinearLayout
62             android:layout_width="match_parent"
63             android:layout_height="wrap_content"
64             android:layout_margin="5dp"
65             android:orientation="horizontal">

```

Menggunakan Cardview.widget agar tampilannya dapat seperti kartu.

● Baris 10-127

Merupakan design agar bisa menampilkan nama, items, address, total price, dan payment methode, yang akan tampil di dalam cardview nya

```

<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Address"
    android:layout_marginLeft="10dp"
    android:textColor="@color/white"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/or_alamat"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_weight="1"
    android:text="Address"
    android:textColor="@color/white"
    android:textStyle="bold"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
    <androidx.appcompat.widget.AppCompatTextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="Total"
        android:layout_marginLeft="10dp"
        android:textColor="@color/white"/>

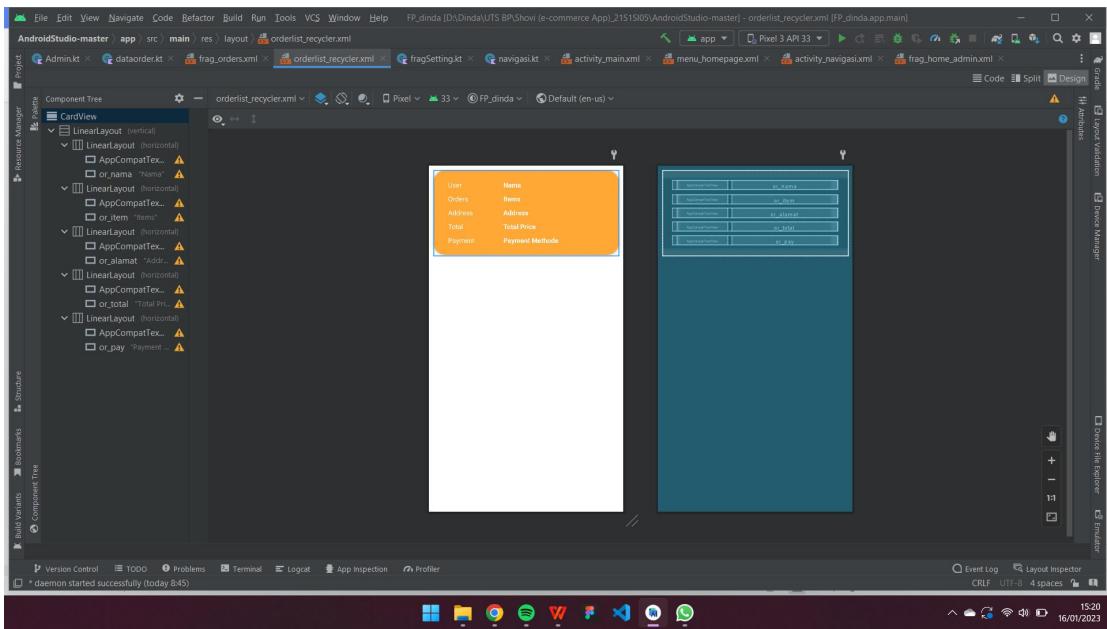
```

```

100     android:text="Total Price"
101     android:textColor="@color/white"
102     android:textStyle="bold"/>
103 </LinearLayout>
104 <LinearLayout
105     android:layout_width="match_parent"
106     android:layout_height="wrap_content"
107     android:layout_margin="5dp"
108     android:orientation="horizontal">
109     <androidx.appcompat.widget.AppCompatTextView
110         android:layout_width="match_parent"
111         android:layout_height="wrap_content"
112         android:layout_weight="2"
113         android:layout_marginLeft="10dp"
114         android:text="Payment"
115         android:textColor="@color/white"/>
116     <androidx.appcompat.widget.AppCompatTextView
117         android:id="@+id/or_pay"
118         android:layout_width="match_parent"
119         android:layout_height="wrap_content"
120         android:layout_weight="1"
121         android:layout_marginLeft="10dp"
122         android:text="Payment Methode"
123         android:textColor="@color/white"
124         android:textStyle="bold"
125     />
126 </LinearLayout>
127 </LinearLayout>
128 </androidx.cardview.widget.CardView>

```

UI



F. row_layout.xml

Layout ini merupakan design dari list database atau data product yang diinputkan secara dinamis.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.cardview.widget.CardView xmlns:android="http://schemas.android.co
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     app:cardBackgroundColor="#ffa836"
7     app:cardCornerRadius="20dp"
8     app:cardElevation="5dp"
9     android:layout_margin="10dp"
10    >
11
12    <LinearLayout
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:orientation="vertical"
16        android:layout_margin="16dp"
17        >
18        <TextView
19            android:layout_width="match_parent"
20            android:layout_height="wrap_content"
21            android:text="Shov1's Product"
22            android:textSize="20dp"
23            android:textStyle="bold"
24            android:layout_marginBottom="5dp"
25            android:layout_marginLeft="10dp"/>
26        <LinearLayout
27            android:layout_width="match_parent"
28            android:layout_height="wrap_content"
29            android:orientation="horizontal"
```

Menggunakan Cardview.widget agar tampilannya dapat seperti kartu.

- Baris 12-120

Merupakan design agar bisa menampilkan ID, nama produk, price, dan stok yang akan tampil di dalam cardview nya

```

<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:layout_marginLeft="10dp"
    android:text="10"
    android:textColor="@color/white"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_row_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_weight="1"
    android:text="ID"
    android:textSize="20dp"
    android:textColor="@color/white"
    android:textStyle="bold"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Nama Produk"
    android:textColor="@color/white"
    android:textStyle="bold"/>
</LinearLayout>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_marginLeft="10dp"
    android:text="Nama Produk"
    android:textSize="20dp"
    android:textColor="@color/white"
    android:textStyle="bold"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Harga"
    android:layout_marginLeft="10dp"
    android:textColor="@color/white"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_harga"
    android:layout_width="match.parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_marginLeft="10dp"/>
</LinearLayout>

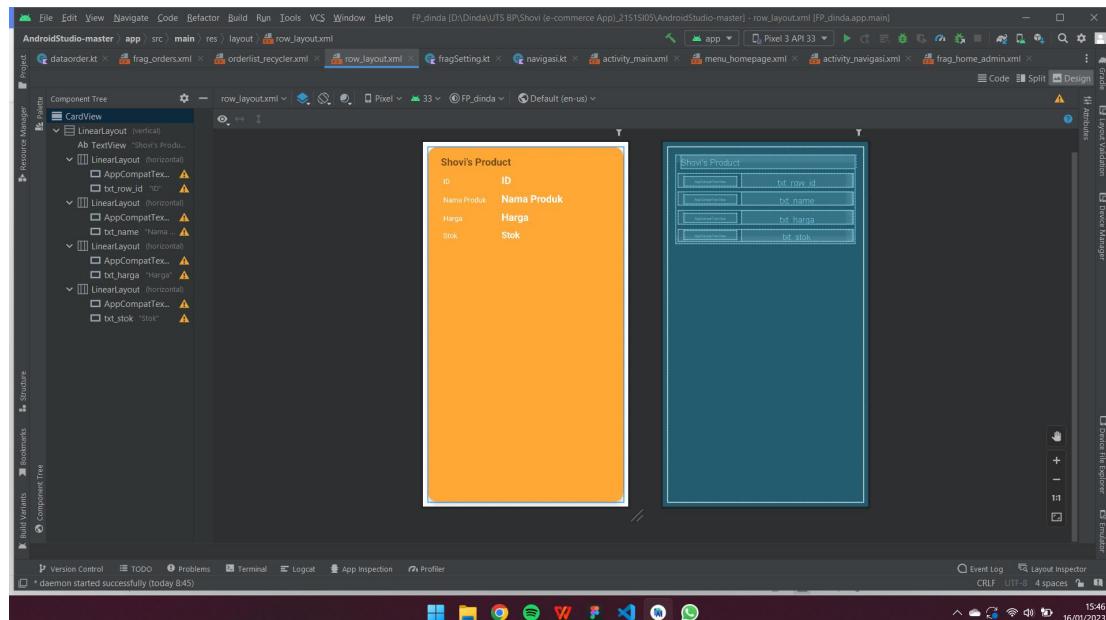
```

```

<CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="ID"
    android:layout_marginLeft="10dp"
    android:textColor="@color/white"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_id"
    android:layout_width="match.parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Nama Produk"
    android:layout_marginLeft="10dp"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_name"
    android:layout_width="match.parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Harga"
    android:layout_marginLeft="10dp"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_harga"
    android:layout_width="match.parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="horizontal">
<androidx.appcompat.widget.AppCompatTextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="Stok"
    android:layout_marginLeft="10dp"/>
<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/txt_stok"
    android:layout_width="match.parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</CardView>

```

UI



G. setting.xml

Layout ini berisikan design pada halaman setting yang juga merupakan fragment.

Halaman ini menyambung dengan file fragSetting.kt

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".fragment.FragSetting">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_marginTop="24dp"
            android:layout_marginLeft="24dp"
            android:layout_marginRight="24dp">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Setting"
                android:textSize="16dp"
                android:textColor="@color/black"/>
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Shovi's Admin"
                android:textSize="28dp"/>
            <Button
                android:id="@+id/setting"
                android:layout_width="match_parent"
                android:layout_height="70dp"
                android:layout_gravity="start"
                android:text="General Setting"
                android:fontFamily="@font/poppins"
                android:textStyle="bold"
                android:textAlignment="textStart"
                android:textAllCaps="false"
                app:backgroundTint="#ffa836" />
            <Button
                android:id="@+id/detail"
                android:layout_width="match_parent"
                android:layout_height="70dp"
                android:layout_gravity="start"
                android:text="General Detail"
                android:fontFamily="@font/poppins"
                android:textStyle="bold"
                android:textAlignment="textStart"
                android:textAllCaps="false"
                app:backgroundTint="#ffa836" />
            <Button
                android:id="@+id/homeadmin"
                android:layout_width="match_parent"
                android:layout_height="70dp"
                android:text="Home Admin"
                android:fontFamily="@font/poppins"
                android:textStyle="bold"
                android:textAlignment="textStart"
                android:textAllCaps="false"
                app:backgroundTint="#ffa836" />
        
    

```

Menggunakan LinearLayout sebagai default layout halaman ini. Lalu halaman ini dibungkus lagi dengan ScrollView secara vertical agar terlihat lebih rapi lagi.

- Baris 8-32

Menggunakan scrollView agar dapat di scroll, Menggunakan tag TextView untuk menuliskan tulisan Setting dan Shovi's Admin sebagai hiasan pada halaman ini.

- Baris 34-44

Menggunakan tag Button untuk membuatnya sebagai button, Memberikan id dengan nama setting, button ini memiliki text yaitu General Setting, namun button ini tidak difungsikan.

- Baris 45-55

Menggunakan tag Button untuk membuatnya sebagai button, Memberikan id dengan nama detail, button ini memiliki text yaitu General Detail, namun button ini tidak difungsikan.

- Baris 57-67

Menggunakan tag Button untuk membuatnya sebagai button, Memberikan id dengan nama homeadmin, button ini memiliki text yaitu Home Admin, dan dapat berfungsi apabila diklik.

The image shows two side-by-side snippets of XML code within the Android Studio interface. Both snippets are under the 'Resource Manager' tab and are titled 'Buttons'. The left snippet contains three buttons with IDs 'order', 'payment', and 'catalog'. The right snippet contains four buttons with IDs 'notif', 'lang', and 'faq'. Each button has specific styling like font family (@font/poppins), bold text, and background tint colors (#e69509 for shipping/payment/catalog and #ffa836 for notification/language/FAQ).

```
<Button  
    android:id="@+id/order"  
    android:layout_width="match_parent"  
    android:layout_height="70dp"  
    android:text="Shipping"  
    android:fontFamily="@font/poppins"  
    android:textStyle="bold"  
    android:textAlignment="textStart"  
    android:textAllCaps="false"  
    android:layout_gravity="start"  
    app:backgroundTint="#e69509"/>  
  
<Button  
    android:id="@+id/payment"  
    android:layout_width="match_parent"  
    android:layout_height="70dp"  
    android:text="Payments"  
    android:fontFamily="@font/poppins"  
    android:textStyle="bold"  
    android:textAlignment="textStart"  
    android:textAllCaps="false"  
    android:layout_gravity="start"  
    app:backgroundTint="#ffa836"/>  
  
<Button  
    android:id="@+id/catalog"  
    android:layout_width="match_parent"  
    android:layout_height="70dp"  
    android:text="Catalog Management"  
    android:textStyle="bold"  
    android:textAlignment="textStart"  
    android:textAllCaps="false"  
    android:layout_gravity="start"  
    android:backgroundTint="#e69509"/>  
  
  
<Button  
    android:id="@+id/notif"  
    android:layout_width="match_parent"  
    android:layout_height="70dp"  
    android:layout_gravity="start"  
    android:layout_marginTop="20dp"  
    android:text="Notification"  
    android:fontFamily="@font/poppins"  
    android:textStyle="bold"  
    android:textAlignment="textStart"  
    android:textAllCaps="false"  
    app:backgroundTint="#ffa836" />  
  
<Button  
    android:id="@+id/lang"  
    android:layout_width="match_parent"  
    android:layout_height="70dp"  
    android:text="Languages"  
    android:fontFamily="@font/poppins"  
    android:textStyle="bold"  
    android:textAlignment="textStart"  
    android:textAllCaps="false"  
    android:layout_gravity="start"  
    app:backgroundTint="#ffa836"/>  
  
<Button  
    android:id="@+id/faq"  
    android:layout_width="match_parent"  
    android:layout_height="70dp"  
    android:text="FAQs"
```

- Baris 68-78

Menggunakan tag Button untuk membuatnya sebagai button, Memberikan id dengan nama order, button ini memiliki text yaitu Shipping, dan dapat berfungsi apabila di klik.

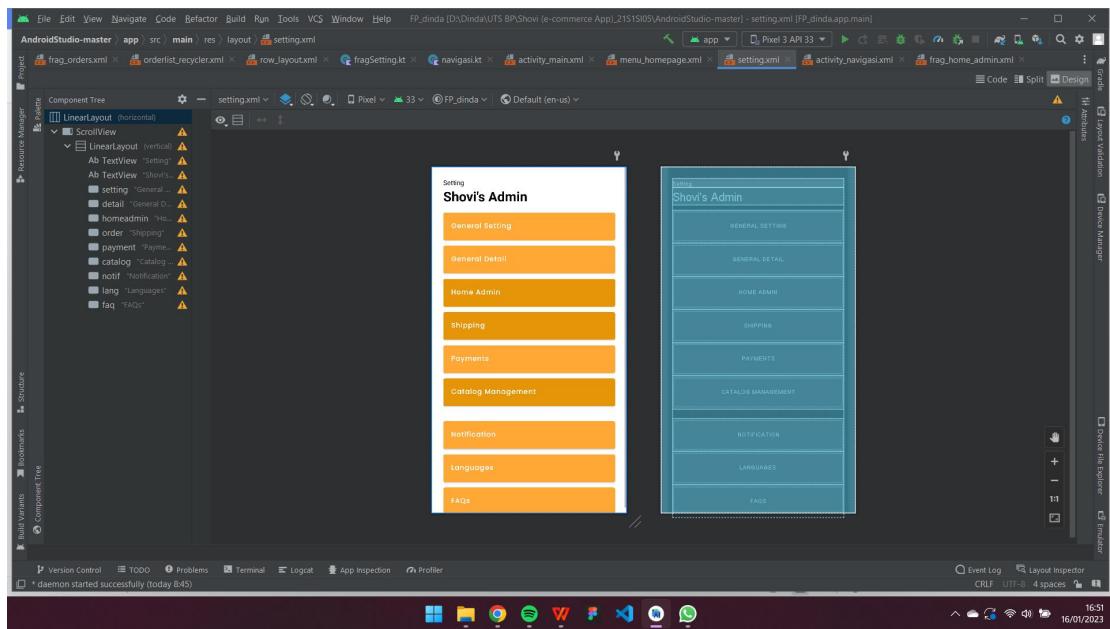
- Baris 79-89

Menggunakan tag Button untuk membuatnya sebagai button, Memberikan id dengan nama payment, button ini memiliki text yaitu Payments, namun button ini tidak difungsikan.

- Baris 90-100

Menggunakan tag Button untuk membuatnya sebagai button, Memberikan id dengan nama catalog, button ini memiliki text yaitu Catalog Management, dan dapat berfungsi apabila di klik.

UI



H. tambah_produk.xml

Layout ini merupakan halaman CRUD ketika produk akan ditambahkan, diedit, atau dihapus.

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6       <LinearLayout
7         android:layout_width="match_parent"
8         android:layout_height="wrap_content"
9         android:orientation="horizontal"
10        android:layout_margin="10dp"
11        android:layout_marginBottom="10dp"
12        android:layout_marginLeft="10dp"
13        android:layout_marginRight="10dp">
14
15         <Button
16           android:id="@+id/btn_back"
17           android:layout_width="match_parent"
18           android:layout_height="wrap_content"
19           android:layout_weight="1"
20           android:text="Back" />
21
22         <TextView
23           android:layout_width="wrap_content"
24           android:layout_height="wrap_content"
25           android:text="Catalog Management"
26           android:layout_gravity="center_horizontal"
27           android:textSize="25sp"
28           android:textStyle="bold"
29           android:layout_margin="10dp"
           android:layout_marginBottom="10dp" />

```

- Menggunakan tag button untuk membuat elemen tersebut menjadi sebuah button. Button ini memiliki id yang kami beri nama btn_back. Terdapat pula tulisan di button ini yaitu Back untuk memberitahu user, jika button ini di klik maka user akan kembali ke halaman home admin

- Menggunakan tag EditText untuk membuat elemen tersebut menjadi sebuah form text yang dapat diisikan oleh user. Form ini kami buat menyesuaikan dengan atribut database yang ada.

```

42
43 <EditText
44     android:id="@+id/edt_id"
45     android:layout_width="match_parent"
46     android:layout_height="wrap_content"
47     android:hint="ID"
48     android:inputType="number"
49     android:layout_margin="10dp"
50     android:layout_marginBottom="10dp"
51     android:layout_marginLeft="10dp"
52     android:layout_marginRight="10dp"
53   />
54 <EditText
55     android:id="@+id/edt_name"
56     android:layout_width="match_parent"
57     android:layout_height="wrap_content"
58     android:inputType="text"
59     android:hint="Nama Produk"
60     android:layout_margin="10dp"
61     android:layout_marginBottom="10dp"
62     android:layout_marginLeft="10dp"
63     android:layout_marginRight="10dp"/>
64 <EditText
65     android:id="@+id/edt_harga"
66     android:layout_width="match_parent"
67     android:layout_height="wrap_content"
68     android:inputType="text"
69     android:hint="Harga"
70     android:layout_margin="10dp"
71
72
73
74
75 <EditText
76     android:id="@+id/edt_stok"
77     android:layout_width="match_parent"
78     android:layout_height="wrap_content"
79     android:inputType="text"
80     android:hint="Stok"
81     android:layout_margin="10dp"
82     android:layout_marginBottom="10dp"
83     android:layout_marginLeft="10dp"
84     android:layout_marginRight="10dp"/>
85 <LinearLayout
86     android:layout_width="match_parent"
87     android:layout_height="wrap_content"
88     android:orientation="horizontal"
89     android:weightSum="3"
90     android:layout_margin="10dp"
91     android:layout_marginBottom="10dp"
92     android:layout_marginLeft="10dp"
93     android:layout_marginRight="10dp"
94     android:gravity="center_horizontal">
95
96   <Button
97     android:id="@+id/btn_add"
98     android:layout_width="wrap_content"
99     android:layout_height="wrap_content"
100    android:layout_weight="1"
101    android:text="Add"
102  />
```

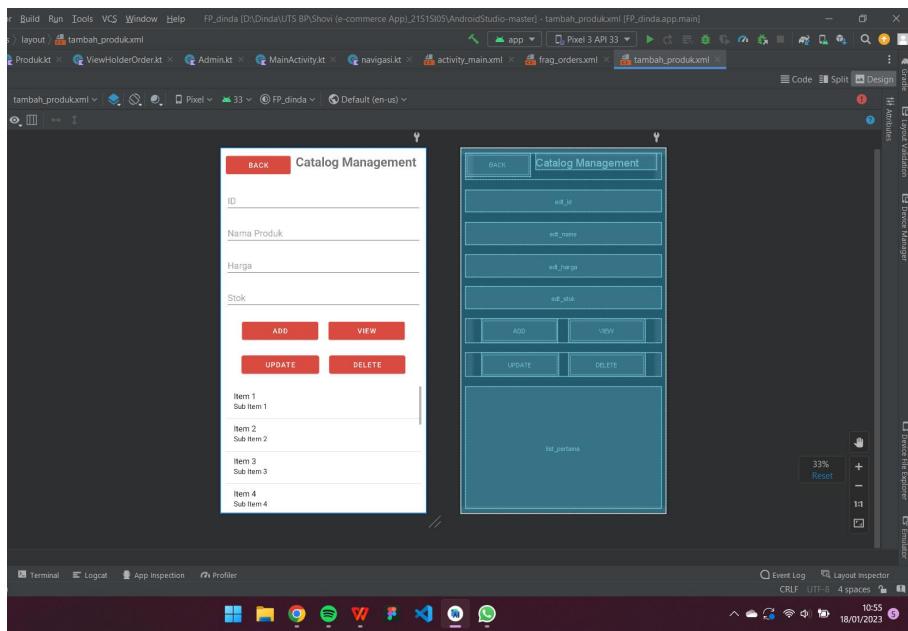
- Menggunakan tag button untuk membuat elemen tersebut menjadi sebuah button. Button ini memiliki id yang kami beri nama sesuai dengan fungsinya masing-masing.seperti apabila mengklik button add maka data catalog yang baru berhasil ditambahkan, lalu apabila mengklik button update, maka update data yang sudah ada berhasil terupdate, lalu apabila mengklik button delete maka data produk yang dipilih akan berhasil terhapus, serta button view yang apabila di klik maka akan muncul semua data catalog yang sudah ada.

```

182      />
183      <Button
184          android:id="@+id/btn_view"
185          android:layout_width="wrap_content"
186          android:layout_height="wrap_content"
187          android:layout_weight="1"
188          android:layout_marginLeft="20dp"
189          android:text="View" />
190
191      </LinearLayout>
192      <LinearLayout
193          android:layout_width="match_parent"
194          android:layout_height="wrap_content"
195          android:orientation="horizontal"
196          android:weightSum="3"
197          android:layout_margin="10dp"
198          android:layout_marginBottom="10dp"
199          android:layout_marginLeft="10dp"
200          android:layout_marginRight="10dp"
201          android:gravity="center_horizontal">
202          <Button
203              android:id="@+id/btn_update"
204              android:layout_width="wrap_content"
205              android:layout_height="wrap_content"
206              android:layout_weight="1"
207              android:text="Update" />
208
209          <Button
210              android:id="@+id/btn_delete"
211              android:layout_width="wrap_content"
212              android:layout_height="wrap_content"
213              android:layout_weight="1"
214              android:text="Delete" />
215
216      </LinearLayout>
217      <ListView
218          android:id="@+id/list_pertama"
219          android:layout_width="match_parent"
220          android:layout_height="wrap_content"
221          android:choiceMode="singleChoice"
222          android:layout_margin="10dp"
223      />
224
225  </LinearLayout>

```

UI



I. menu_homepage.xml

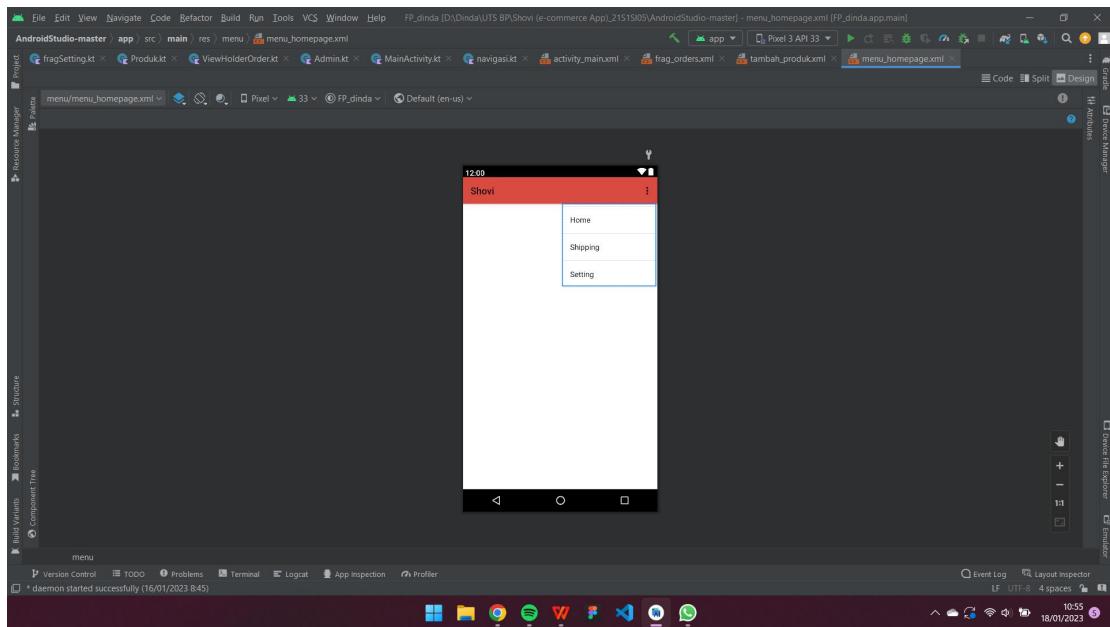
```

<menu 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"
    tools:context="com.example.fp_dinda.homepage">
    <item
        android:id="@+id/homepage"
        android:icon="@drawable/ic_baseline_home_24"
        android:title="Home"
    />
    <item
        android:id="@+id/order"
        android:icon="@drawable/ic_baseline_local_shipping_24"
        android:title="Shipping"
    />
    <item
        android:id="@+id/setting"
        android:icon="@drawable/ic_baseline_settings_24"
        android:title="Setting"
    />
</menu>

```

- Layout ini berguna untuk mengatur desain dari navigasinya.

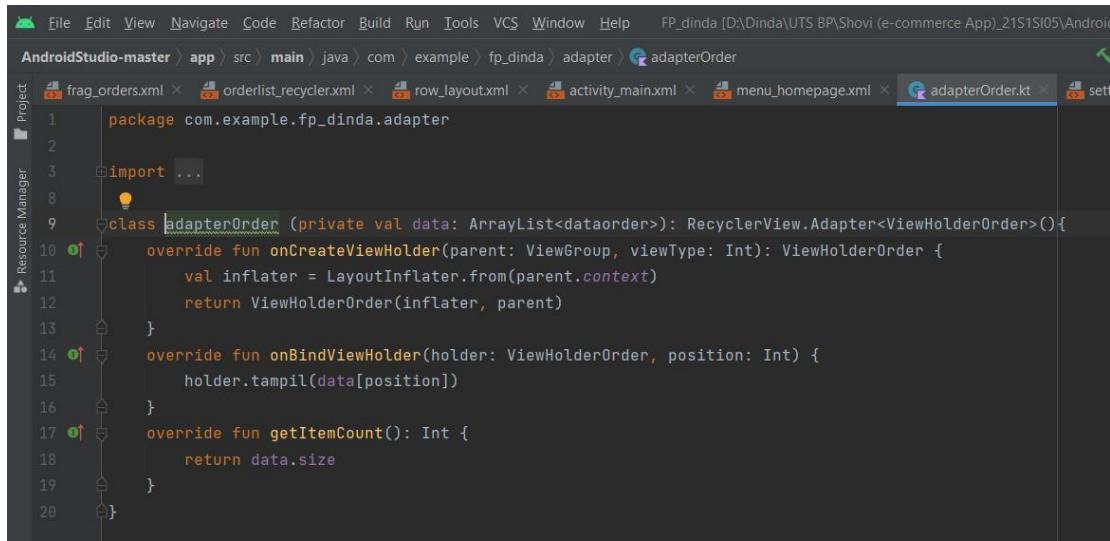
UI



BAB III

PENJELASAN CLASS

A. adapterOrder

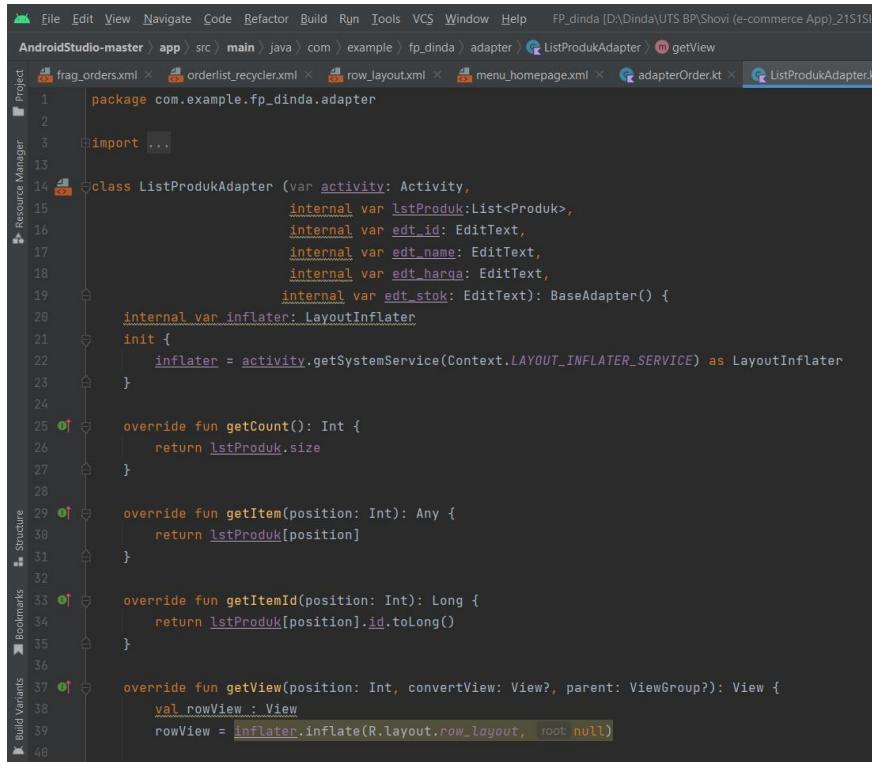


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help FP_dinda [D:\Dinda\UTS BP\Shovi (e-commerce App)_21S1SI05\Android Studio-master] app src main java com example fp_dinda adapter adapterOrder.kt
```

```
1 package com.example.fp_dinda.adapter
2
3 import ...
4
5 class AdapterOrder (private val data: ArrayList<dataorder>): RecyclerView.Adapter<ViewHolderOrder>(){
6     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolderOrder {
7         val inflator = LayoutInflater.from(parent.context)
8         return ViewHolderOrder(inflator, parent)
9     }
10    override fun onBindViewHolder(holder: ViewHolderOrder, position: Int) {
11        holder.tampil(data[position])
12    }
13    override fun getItemCount(): Int {
14        return data.size
15    }
16}
17
18}
```

Penjelasan : class adapterOrder digunakan untuk menyambungkan antara dataorder.kt yang berisi data variable yang digunakan untuk data order dengan ViewHolderOlder.kt yang berfungsi untuk menyambungkan antara dataorder dengan recycler yang digunakan

B. ListProdukAdapter



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help FP_dinda [D:\Dinda\UTS BP\Shovi (e-commerce App)_21S1SI05\Android Studio-master] app src main java com example fp_dinda adapter ListProdukAdapter.kt
```

```
1 package com.example.fp_dinda.adapter
2
3 import ...
4
5 class ListProdukAdapter (var activity: Activity,
6                         internal var lstProduk: List<Produk>,
7                         internal var edt_id: EditText,
8                         internal var edt_name: EditText,
9                         internal var edt_harga: EditText,
10                        internal var edt_stok: EditText): BaseAdapter() {
11    internal var inflator: LayoutInflater
12    init {
13        inflator = activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater
14    }
15
16    override fun getCount(): Int {
17        return lstProduk.size
18    }
19
20    override fun getItem(position: Int): Any {
21        return lstProduk[position]
22    }
23
24    override fun getItemId(position: Int): Long {
25        return lstProduk[position].id.toLong()
26    }
27
28    override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {
29        val rowView : View
30        rowView = inflator.inflate(R.layout.row_layout, root null)
31    }
32
33    override fun getLayoutResource(position: Int): Int {
34        return R.layout.row_layout
35    }
36
37    override fun getLayoutResource(convertView: View, parent: ViewGroup?, position: Int): Int {
38        return R.layout.row_layout
39    }
40}
```

```
28
29     override fun getItem(position: Int): Any {
30         return lstProduk[position]
31     }
32
33     override fun getItemId(position: Int): Long {
34         return lstProduk[position].id.toLong()
35     }
36
37     override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {
38         val rowView : View
39         rowView = inflater.inflate(R.layout.row_layout, root: null)
40
41         rowView.txt_row_id.text = lstProduk[position].id.toString()
42         rowView.txt_name.text = lstProduk[position].name.toString()
43         rowView.txt_harga.text = lstProduk[position].harga.toString()
44         rowView.txt_stok.text = lstProduk[position].stok.toString()
45
46
47         rowView.setOnClickListener{ it:View{
48             edt_id.setText(rowView.txt_row_id.text.toString())
49             edt_name.setText(rowView.txt_name.text.toString())
50             edt_harga.setText(rowView.txt_harga.text.toString())
51             edt_stok.setText(rowView.txt_stok.text.toString())
52         }
53     }
54
55     return rowView
56 }
```

Penjelasan : class ini digunakan untuk menyambungkan antara halaman layout row_layout dengan model Produk.kt yang sudah dibuat

C. dataorder

```
1 package com.example.fp_dinda.data
2
3 class DataOrder ( val nama : String,
4                   val item: Int,
5                   val alamat: String,
6                   val total: Int,
7                   val payment: String)
```

Penjelasan : class ini berguna untuk membuat atau mendeklarasikan variable dari dataorder yang akan diisi pada recycler nantinya, beserta dengan tipe datanya.

D. DBHelper

The image shows three vertically stacked code editor panes from the Android Studio IDE, all displaying the same Java file: `DBHelper.java`. The file is part of the `com.example.fp_dinda` package and defines a database helper class for a `Produk` table.

```
1 package com.example.fp_dinda.DBHelper
2
3 import ...
4
5
6 class DBHelper (context: Context): SQLiteOpenHelper(context, DATABASE_NAME, factory: null, DATABASE_VER) {
7
8     companion object{
9         private val DATABASE_VER = 1
10        private val DATABASE_NAME = "Produk.db"
11
12        private val TABLE_NAME = "Produk"
13        private val COL_ID = "Id"
14        private val COL_NAME = "Name"
15        private val COL_HARGA = "Harga"
16        private val COL_STOK = "Stok"
17    }
18
19    override fun onCreate(db: SQLiteDatabase?) {
20        val CREATE_TABLE_QUERY: String = ("CREATE TABLE $TABLE_NAME($COL_ID INTEGER PRIMARY KEY,$COL_NAME TEXT,$COL_HARGA TEXT,$COL_STOK TEXT)")
21        db!!.execSQL(CREATE_TABLE_QUERY);
22    }
23
24    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
25        db.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
26        onCreate(db!!)
27    }
28
29    val allProduk: List<Produk>
30        get(){
31            val lstProduk = ArrayList<Produk>()
32
33            val lstProduk = ArrayList<Produk>()
34            val selectQuery = "SELECT * FROM $TABLE_NAME"
35            val db: SQLiteDatabase = this.writableDatabase
36            val cursor = db.rawQuery(selectQuery, selectionArgs: null)
37            if(cursor.moveToFirst()){
38                do{
39                    val produk = Produk()
40                    produk.id = cursor.getInt(cursor.getColumnIndexOrThrow(COL_ID))
41                    produk.name = cursor.getString(cursor.getColumnIndexOrThrow(COL_NAME))
42                    produk.harga = cursor.getString(cursor.getColumnIndexOrThrow(COL_HARGA))
43                    produk.stok = cursor.getString(cursor.getColumnIndexOrThrow(COL_STOK))
44
45
46                    lstProduk.add(produk)
47                }while (cursor.moveToNext())
48            }
49            db.close()
50            return lstProduk
51        }
52
53        fun addProduk(produk: Produk){
54            val db = this.writableDatabase
55            val values = ContentValues()
56            values.put(COL_ID, produk.id)
57            values.put(COL_NAME, produk.name)
58            values.put(COL_HARGA, produk.harga)
59            values.put(COL_STOK, produk.stok)
60
61            db.insert(TABLE_NAME, nullColumnHack: null,values)
62            db.close()
63        }
64
65
66        values.put(COL_STOK, produk.stok)
67
68        db.insert(TABLE_NAME, nullColumnHack: null,values)
69        db.close()
70
71        fun updateProduk(produk: Produk): Int{
72            val db = this.writableDatabase
73            val values = ContentValues()
74            values.put(COL_ID, produk.id)
75            values.put(COL_NAME, produk.name)
76            values.put(COL_HARGA, produk.harga)
77            values.put(COL_STOK, produk.stok)
78
79            return db.update(TABLE_NAME, values, whereClause: "$COL_ID=?", arrayOf(produk.id.toString()))
80        }
81
82        fun deleteProduk(produk: Produk) {
83            val db = this.writableDatabase
84
85            db.delete(TABLE_NAME, whereClause: "$COL_ID=?", arrayOf(produk.id.toString()))
86            db.close()
87
88        }
89
90        fun readProduk (): Cursor? {
91            val db = this.readableDatabase
92            return db.rawQuery( sql: "SELECT * FROM " + TABLE_NAME, selectionArgs: null)
93        }
94
95
96
97
98
99 }
```

Penjelasan : Class ini digunakan untuk membuat database beserta dengan table-table nya. Pada class ini pula kami mmebuat CRUD atau fitur menambah, mengupdate, membaca ataupun menghapus data.

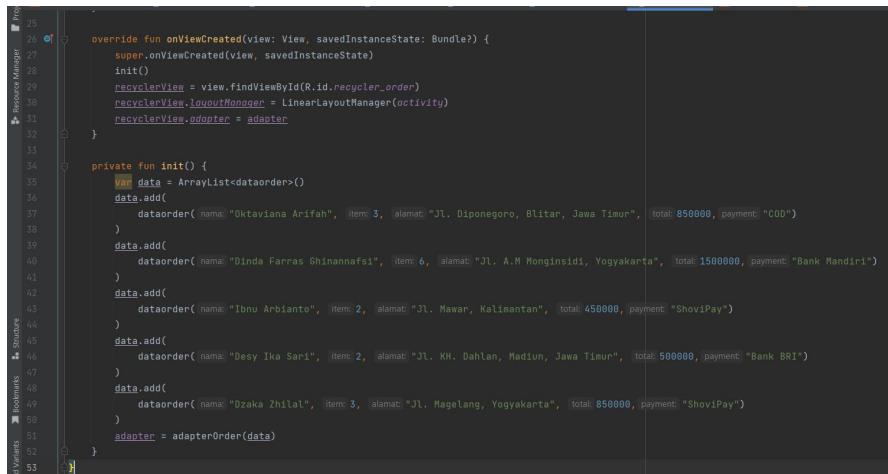
E. fragHomeAdmin

```
1 package com.example.fp_dinda.fragment
2
3 import ...
11
12 class fragHomeAdmin : Fragment() {
13
14     override fun onCreateView(
15         inflater: LayoutInflater, container: ViewGroup?,
16         savedInstanceState: Bundle?
17     ): View? {
18         val bind = FragHomeAdminBinding.inflate(inflater)
19
20         bind.tes1.setOnClickListener{ it:View!
21             val intent = Intent (this@fragHomeAdmin.requireContext(), Admin::class.java)
22             startActivity(intent)
23         }
24         return bind.root
25     }
26
27 }
```

Penjelasan : Class ini merupakan fragment dari halaman homeadmin yang menyambung dengan layout frag_home_Admin.

F. fragorders

```
1 package com.example.fp_dinda.fragment
2
3 import ...
15
16 class fragorders : Fragment() {
17
18     lateinit var recyclerView: RecyclerView
19     lateinit var adapter: adapterOrder
20
21     override fun onCreateView(
22         inflater: LayoutInflater, container: ViewGroup?,
23         savedInstanceState: Bundle?
24     ): View? {
25         return inflater.inflate(R.layout.frag_orders, container, false)
26     }
27
28     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
29         super.onViewCreated(view, savedInstanceState)
30         init()
31     }
32
33     private fun init() {
34         var data = ArrayList<dataorder>()
35         data.add(
36             dataorder( name: "Oktaviana Arifah", item: 3, alamat: "Jl. Diponegoro, Blitar, Jawa Timur", total: 850000, payment: "COD")
37         )
38         data.add(
39             dataorder( name: "Dinda Farras Ghinannafsi", item: 6, alamat: "Jl. A.M. Monginsidi, Yogyakarta", total: 1500000, payment: "Bank Mandiri")
```

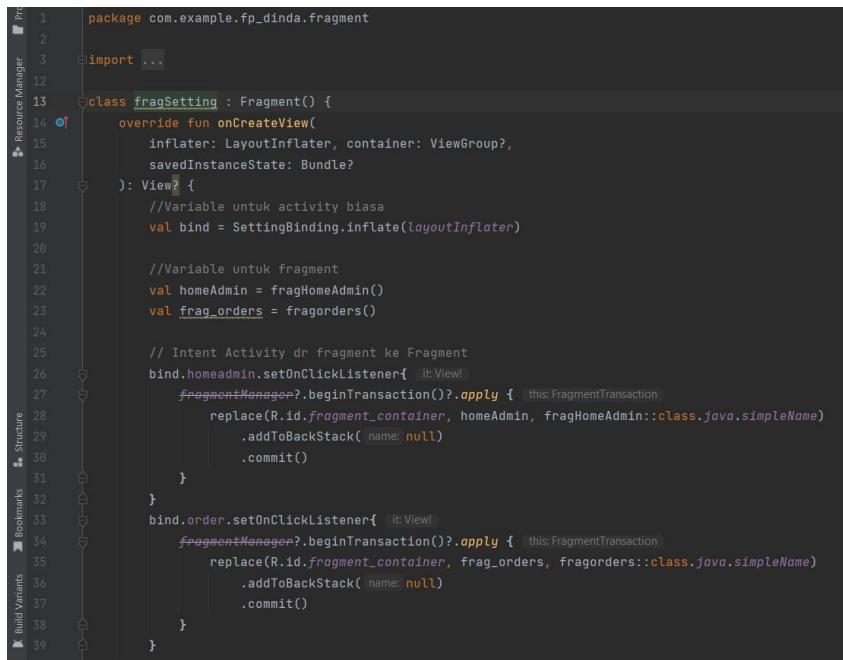


```
override fun onViewCreated(view: View, savedInstanceState: Bundle) {
    super.onViewCreated(view, savedInstanceState)
    init()
    recycler_view = view.findViewById(R.id.recycler_order)
    recycler_view.layoutManager = LinearLayoutManager(activity)
    recycler_view.adapter = adapter
}

private fun init() {
    var data = ArrayList<dataorder>()
    data.add(
        dataorder( name: "Oktaviana Arifah", item: 3, alamat: "Jl. Diponegoro, Blitar, Jawa Timur", total: 850000, payment: "COD")
    )
    data.add(
        dataorder( name: "Dinda Farras Ghinannafsi", item: 6, alamat: "Jl. A.M Monginsidi, Yogyakarta", total: 1500000, payment: "Bank Mandiri")
    )
    data.add(
        dataorder( name: "Ibnu Arbianto", item: 2, alamat: "Jl. Mawar, Kalimantan", total: 450000, payment: "ShoviPay")
    )
    data.add(
        dataorder( name: "Desy Ika Sari", item: 2, alamat: "Jl. KH. Dahlan, Madiun, Jawa Timur", total: 500000, payment: "Bank BRI")
    )
    data.add(
        dataorder( name: "Dzaka Zhilal", item: 3, alamat: "Jl. Magelang, Yogyakarta", total: 850000, payment: "ShoviPay")
    )
    adapter = adapterOrder(data)
}
```

Penjelasan : Class ini merupakan fragment dari halaman shipping atau order, dimana halaman tersebut berisi data recycler. Sehingga, pada class ini pula kami mengisikan data pada recyclernya.

G. fragSetting



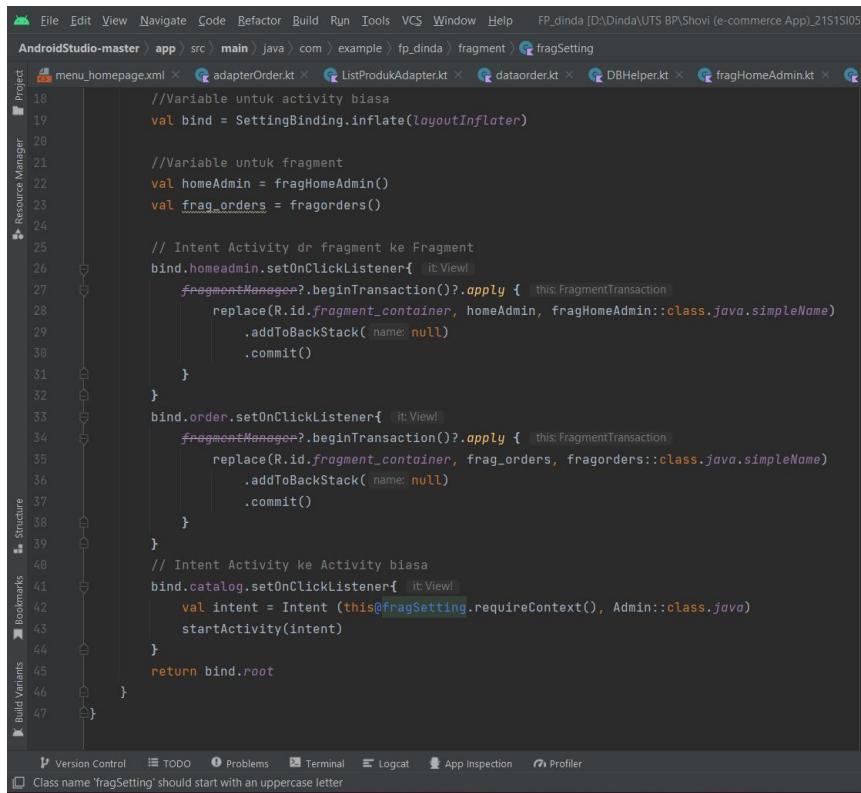
```
package com.example.fp_dinda.fragment

import ...

class fragSetting : Fragment() {
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        //Variable untuk activity biasa
        val bind = SettingBinding.inflate(inflater)

        //Variable untuk fragment
        val homeAdmin = fragHomeAdmin()
        val frag_orders = fragorders()

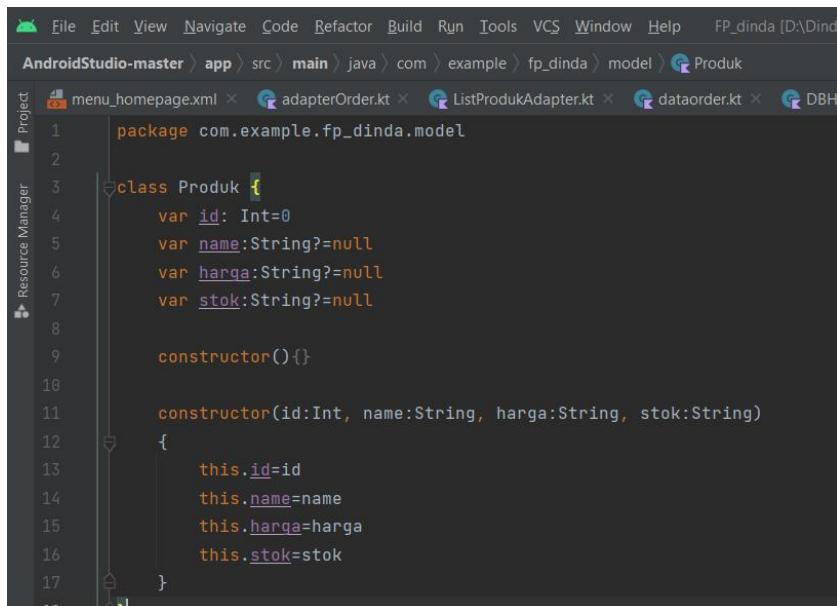
        // Intent Activity dr fragment ke Fragment
        bind.homeadmin.setOnClickListener { it: View! }
            fragmentManager?.beginTransaction()?.apply { this.FragmentTransaction.replace(R.id.fragment_container, homeAdmin, fragHomeAdmin::class.java.simpleName)
                .addToBackStack( name: null)
                .commit()
            }
        }
        bind.order.setOnClickListener { it: View!
            fragmentManager?.beginTransaction()?.apply { this.FragmentTransaction.replace(R.id.fragment_container, frag_orders, fragorders::class.java.simpleName)
                .addToBackStack( name: null)
                .commit()
            }
        }
    }
}
```



```
18 //Variable untuk activity biasa
19 val bind = SettingBinding.inflate(layoutInflater)
20
21 //Variable untuk fragment
22 val homeAdmin = fragHomeAdmin()
23 val frag_orders = fragOrders()
24
25 // Intent Activity dr fragment ke Fragment
26 bind.homeadmin.setOnClickListener{ it:View!
27     fragmentManager?.beginTransaction()?.apply { this:FragmentTransaction
28         replace(R.id.fragment_container, homeAdmin, fragHomeAdmin::class.java.simpleName)
29         .addToBackStack( name: null )
30         .commit()
31     }
32 }
33 bind.order.setOnClickListener{ it:View!
34     fragmentManager?.beginTransaction()?.apply { this:FragmentTransaction
35         replace(R.id.fragment_container, frag_orders, fragorders::class.java.simpleName)
36         .addToBackStack( name: null )
37         .commit()
38     }
39 }
40 // Intent Activity ke Activity biasa
41 bind.catalog.setOnClickListener{ it:View!
42     val intent = Intent (this@fragSetting.requireContext(), Admin::class.java)
43     startActivity(intent)
44 }
45 return bind.root
46 }
47 }
```

Penjelasan : class ini merupakan fragment dari halaman setting, dimana kita mengaktifkan button yang kami butuhkan di halaman ini, seperti button dengan id homeadmin akan menuju fragment home admin jika di klik, lalu button dengan id order akan menuju ke halaman shipping atau fragorders apabila di klik, serta button dengan id catalog akan menuju ke halaman crud catalog apabila di klik. Class ini menyambung dengan layout dengan nama setting.xml

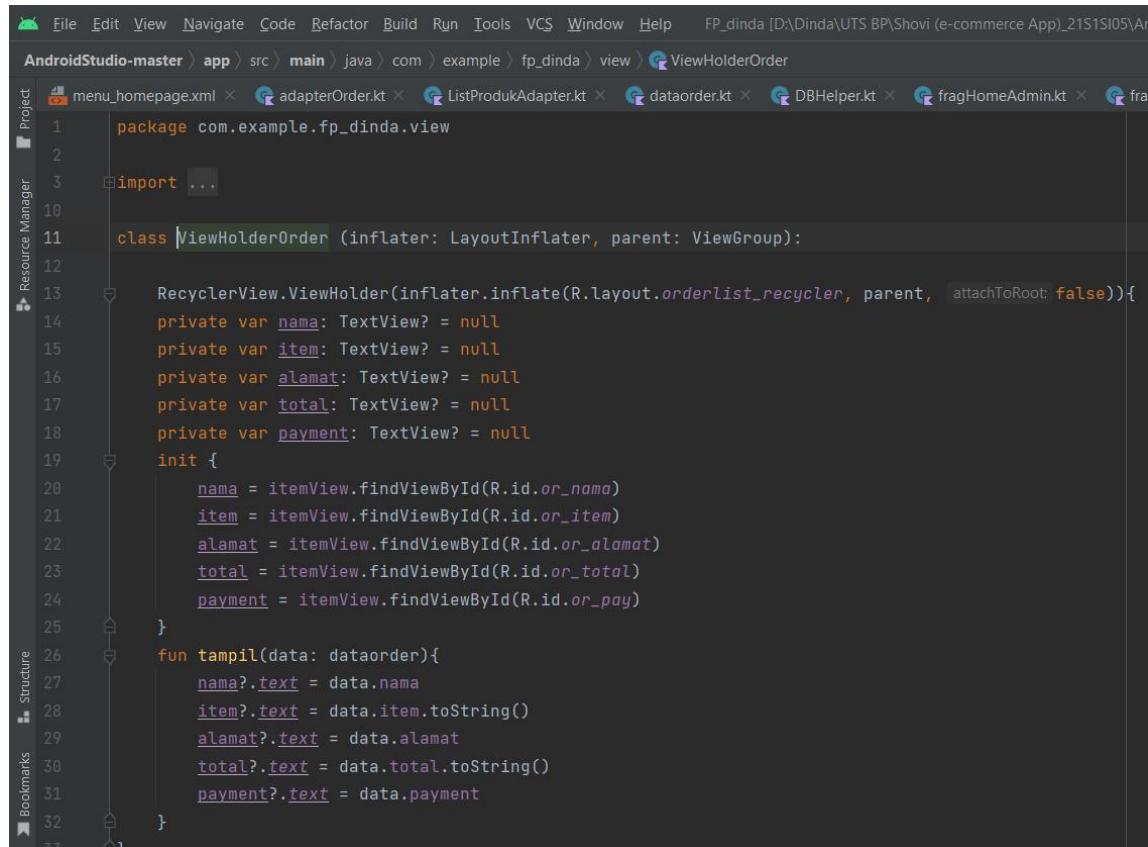
H. Produk



```
1 package com.example.fp_dinda.model
2
3 class Produk {
4     var id: Int=0
5     var name:String?=null
6     var harga:String?=null
7     var stok:String?=null
8
9     constructor(){}
10
11    constructor(id:Int, name:String, harga:String, stok:String)
12    {
13        this.id=id
14        this.name=name
15        this.harga=harga
16        this.stok=stok
17    }
18 }
```

Penjelasan : class ini berguna untuk membuat atau mendeklarasikan variabel yang dibutuhkan pada database, beserta dengan tipe datanya.

I. ViewHolderOrder



The screenshot shows the Android Studio interface with the code editor open to the `ViewHolderOrder.kt` file. The file contains Java code for a ViewHolder class used in a RecyclerView. The code includes imports, a constructor that inflates a layout, initializes views by ID, and a `tampil` function that sets text for five TextViews based on a `dataorder` object. The code editor has syntax highlighting and auto-completion features.

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help FP_dinda [D:\Dinda\UTS BP\Shovi (e-commerce App)_21S1SI05\Ar
AndroidStudio-master > app > src > main > java > com > example > fp_dinda > view > ViewHolderOrder
menu_homepage.xml × adapterOrder.kt × ListProdukAdapter.kt × dataorder.kt × DBHelper.kt × fragHomeAdmin.kt × frag
Project Resource Manager
1 package com.example.fp_dinda.view
2
3 import ...
4
5 class ViewHolderOrder (inflater: LayoutInflater, parent: ViewGroup):
6     RecyclerView.ViewHolder(inflater.inflate(R.layout.orderlist_recycler, parent, attachToRoot: false)){
7     private var nama: TextView? = null
8     private var item: TextView? = null
9     private var alamat: TextView? = null
10    private var total: TextView? = null
11    private var payment: TextView? = null
12    init {
13        nama = itemView.findViewById(R.id.or_nama)
14        item = itemView.findViewById(R.id.or_item)
15        alamat = itemView.findViewById(R.id.or_alamat)
16        total = itemView.findViewById(R.id.or_total)
17        payment = itemView.findViewById(R.id.or_pay)
18    }
19    fun tampil(data: dataorder){
20        nama?.text = data.nama
21        item?.text = data.item.toString()
22        alamat?.text = data.alamat
23        total?.text = data.total.toString()
24        payment?.text = data.payment
25    }
26}
27
28
29
30
31
32}
```

Penjelasan : Class ini berguna untuk menyambungkan antara `dataorder.kt` yang berisikan variabel yang dibutuhkan pada recycler dengan `orderlist_recycler.xml` yang merupakan recycler dedari data yang ada atau recycler yang nantinya ditampilkan di halaman shipping.

J. Admin

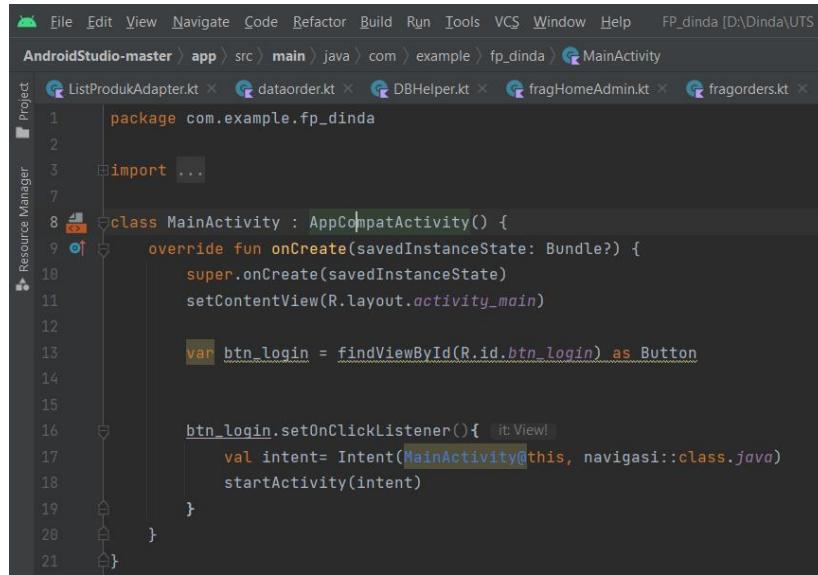
```

1 package com.example.fp_dinda
2
3 import ...
4
5
6 class Admin : AppCompatActivity() {
7     internal lateinit var db: DBHelper
8     internal var lstProduks: List<Produk> = ArrayList<Produk>()
9
10
11
12
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.tambah_produk)
16         db = DBHelper(context = this)
17
18
19         btn_back.setOnClickListener { it: View! }
20             val intent= Intent(MainActivity@this, fragHomeAdmin::class.java)
21             startActivity(intent)
22
23
24         btn_add.setOnClickListener { it: View!
25             val produk = Produk(
26                 Integer.parseInt(edt_id.text.toString()),
27                 edt_name.text.toString(),
28                 edt_harga.text.toString(),
29                 edt_stok.text.toString()
30             )
31             db.addProduk(produk)
32             refreshData()
33
34
35
36
37
38     }
39
40         btn_update.setOnClickListener { it: View!
41             val produk = Produk(
42                 Integer.parseInt(edt_id.text.toString()),
43                 edt_name.text.toString(),
44                 edt_harga.text.toString(),
45                 edt_stok.text.toString()
46             )
47             db.updateProduk(produk)
48             refreshData()
49
50         btn_delete.setOnClickListener { it: View!
51             val produk = Produk(
52                 Integer.parseInt(edt_id.text.toString()),
53                 edt_name.text.toString(),
54                 edt_harga.text.toString(),
55                 edt_stok.text.toString()
56             )
57             db.deleteProduk(produk)
58             refreshData()
59
60         btn_view.setOnClickListener { it: View!
61             db.readProduk()
62             refreshData()
63
64     }
65
66     private fun refreshData(){
67         lstProduks = db.allProduk
68         val adapter = ListProdukAdapter(activity = this@Admin, lstProduks, edt_id, edt_name, edt_harga, edt_stok)

```

Penjelasan : class ini merupakan class dari halaman CRUD catalog baju, dimana kita dapat mengatur button yang terdapat pada halaman CRUD berfungsi sesuai yang smeestinya, seperti apabila mengklik button add maka data catalog yang baru berhasil ditambahkan, lalu apabila mengklik button update, maka update data yang sudah ada berhasil terupdate, lalu apabila mengklik button delete maka data produk yang dipilih akan berhasil terhapus, serta button view yang apabila di klik maka akan muncul semua data catalog yang sudah ada.

K. MainActivity



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help FP_dinda [D:\Dinda\UTS BP\Showi (e-commerce App)_21515]

AndroidStudio-master > app > src > main > java > com > example > fp_dinda > MainActivity.kt

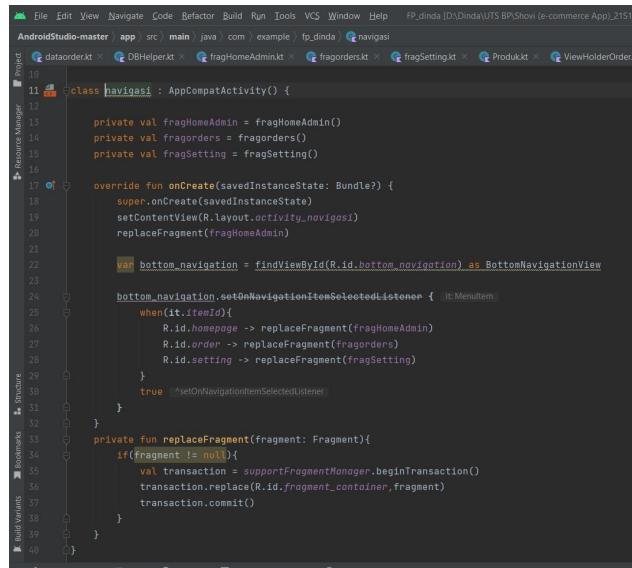
Resource Manager
Project

1 package com.example.fp_dinda
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_main)
9
10        var btn_login = findViewById(R.id.btn_login) as Button
11
12        btn_login.setOnClickListener{ it:View ->
13            val intent= Intent(MainActivity.this, navigasi::class.java)
14            startActivity(intent)
15        }
16    }
17
18 }

19
20
21 }
```

Penjelasan : class ini merupakan class dari halaman utama aplikasi yang kami buat. Class ini menyambung dengan layout dengan nama activity_main.xml. Ada satu button yang berfungsi dihalaman ini yaitu button dengan id btn_login, yang mana apabila button ini di klik maka akan menuju ke halaman home admin.

L. navigasi



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help FP_dinda [D:\Dinda\UTS BP\Showi (e-commerce App)_21515]

AndroidStudio-master > app > src > main > java > com > example > fp_dinda > navigasi.kt

Resource Manager
Project

10
11 class navigasi : AppCompatActivity() {
12
13     private val fragHomeAdmin = fragHomeAdmin()
14     private val fragorders = fragorders()
15     private val fragSetting = fragSetting()
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         setContentView(R.layout.activity_navigasi)
20         replaceFragment(fragHomeAdmin)
21
22         var bottom_navigation = findViewById(R.id.bottom_navigation) as BottomNavigationView
23
24         bottom_navigation.setOnNavigationItemSelectedListener { item:MenuItem ->
25             when(item.itemId){
26                 R.id.homepage -> replaceFragment(fragHomeAdmin)
27                 R.id.order -> replaceFragment(fragorders)
28                 R.id.setting -> replaceFragment(fragSetting)
29             }
30             true
31         }
32     }
33
34     private fun replaceFragment(fragment: Fragment){
35         if(fragment != null){
36             val transaction = supportFragmentManager.beginTransaction()
37             transaction.replace(R.id.fragment_container,fragment)
38             transaction.commit()
39         }
40     }
41 }
```

Penjelasan : class ini merupakan class navigasi yang mana pada class ini kami mengaktifkan semua navigasi yang ada agar menyambung dengan halaman atau fragment yang ingin dituju.

BAB IV
JOBDESK

Nama	NIM	Jobdesk
Dinda Farras Ghinannafsi	21.12.2170	<p>Membuat Laporan Lengkap Pada Fitur Home Admin</p> <ul style="list-style-type: none"> ● Membuat desain pada file fraghomeadmin.xml ● Menyambungkan Button di halaman home admin ke halaman catalog <p>Pada Fitur Catalog</p> <ul style="list-style-type: none"> ● Membuat database ● Membuat desain pada catalog ● Mengaktifkan button yang ada Navigasi ● Menyambungkan semua fragment yang ada agar dapat digunakan di navigasi
Oktaviana Arifah W	21.12.2147	<p>Pada Fitur Shipping</p> <ul style="list-style-type: none"> ● Bekerja sama dengan Desy Ika dalam membuat Viewholder, Adapter, dan data order ● Membuat desain recycler.xml yang akan digunakan di halaman shipping
Desy Ikasari	21.12.2141	<p>Pada Fitur Shipping,</p> <ul style="list-style-type: none"> ● Mengisikan data pada recycler di file fragorders.kt ● Mendesain fragorders.xml ● Bekerja sama dengan Octaviana membuat Adapter,

		<ul style="list-style-type: none"> ● View Holder dan dataorder
Ibnu Arbianto	21.12.2143	<p>Pada Fitur Setting,</p> <ul style="list-style-type: none"> ● Menyambungkan button yang ada di halaman setting ke halaman home admin, shipping, dan catalog di file fragsetting.kt ● Membantu Dzaka Zhilal dalam membuat desain fragsetting.xml
Dzaka Zhilal	21.12.2146	<p>Pada Fitur Setting,</p> <ul style="list-style-type: none"> ● Membuat desain fragsetting.xml ● Membantu Ibnu Arbianto, menyambungkan button di file fragsetting.kt

BAB V
DAFTAR PUSTAKA

Modul Praktikum dan Teori Matakuliah Bahasa Pemograman 2, Universitas Amikom
Yogyakarta.