

PRAKTIKUM MCS BAB 2

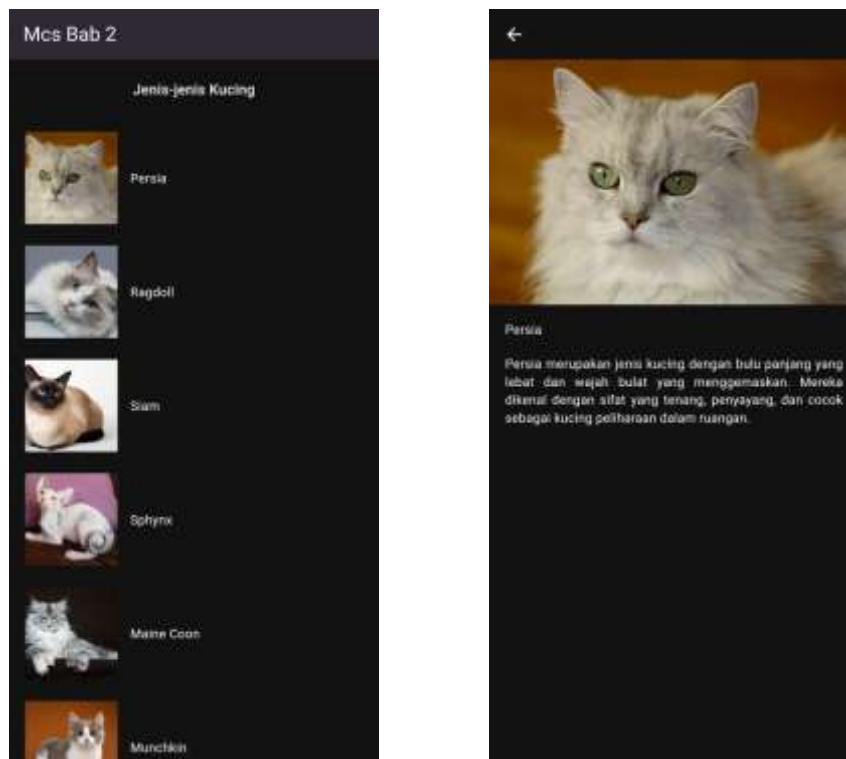
Dinamis UI

PENDAHULUAN

Pada praktikum bab 2 akan membuat ui secara dinamis menggunakan ListView.builder. UI pada aplikasi akan dibuat secara dinamis berdasarkan data yang tersedia. Gambarnya adalah jika ListView() pada bab 1 berisi widget-widget yang dibuat secara manual maka di bab ini akan mengisi widget-widget di dalam ListView() dibentuk secara otomatis sesuai dengan data yang tersedia.

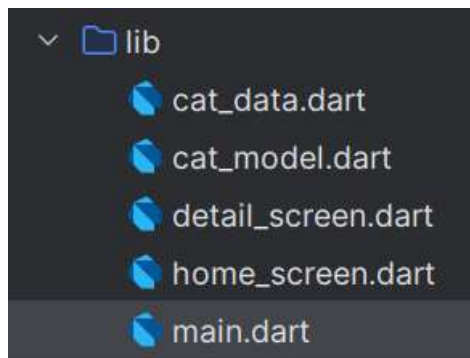
PRAKTIKUM BAB 2

Tampilan aplikasi yang akan dibangun



Penjelasan cara kerja aplikasi akan dijelaskan oleh Penanggung Jawab (PJ).

Tambahkan package google_fonts: ke pubspec.yaml untuk merubah font dari aplikasi dan buat beberapa file .dart seperti di bawah :



Buka file `cat_model.dart` dan berikan isi dengan code sebagai berikut :

```
class CatModel{
  String urlImage;
  String name;
  String desc;

  CatModel({
    required this.urlImage,
    required this.name,
    required this.desc,
  });
}
```

Kita membuat class bernama `CatModel()` beserta constructor di dalamnya. Class `CatModel()` digunakan sebagai kerangka sebuah data yang memiliki ketentuan dari suatu program. Untuk kegunaan detailnya akan terlihat setelah selesai membuat data dummy.

Buka file bernama `cat_data.dart` dan berikan code di dalamnya sebagai berikut :

```
import 'package:mcs_bab_2/cat_model.dart';

List<CatModel> cats = [
  CatModel(
    urlImage:
      "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/persia.jpg",
    name: "Persia",
    desc:
      "'Persia merupakan jenis kucing dengan bulu panjang yang lebat dan
wajah bulat yang menggemaskan. Mereka dikenal dengan sifat yang tenang,
penyayang, dan cocok sebagai kucing peliharaan dalam ruangan.'"),
  CatModel(
    urlImage:
      "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/ragdoll.png",
    name: "Ragdoll",
    desc:
      "'Ragdoll merupakan kucing besar dengan bulu panjang yang halus dan
```

```

mata biru memikat. Mereka terkenal dengan kecenderungan mereka untuk rileks
dan melonggar saat diangkat, mirip dengan boneka ragdoll, dan sangat
penyayang.''),
    CatModel(
        urlImage:
            "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/siam.jpeg",
        name: "Siam",
        desc:
            "'Siam merupakan kucing dengan bulu pendek, mata biru tajam, dan
tubuh yang ramping. Mereka dikenal sebagai kucing vokal yang suka berbicara
dan memiliki kepribadian yang aktif serta ramah.'"),
    CatModel(
        urlImage:
            "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/sphynx.jpg",
        name: "Sphynx",
        desc:
            "'Sphynx merupakan jenis kucing tanpa bulu yang memiliki kulit
lunak seperti kulit jeruk. Mereka sering menjadi perhatian dengan
penampilan yang unik dan ramah serta cerdas dalam perilaku mereka.'"),
    CatModel(
        urlImage:
            "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/maine%20coon.jpg",
        name: "Maine Coon",
        desc:
            "'Maine Coon merupakan salah satu kucing terbesar dengan bulu
panjang dan ekor berbulu tebal. Mereka memiliki sifat yang ramah, lunak,
dan cenderung energik, serta memiliki fisik yang kuat.'"),
    CatModel(
        urlImage:
            "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/munchkin.jpg",
        name: "Munchkin",
        desc:
            "'Munchkin merupakan jenis kucing dengan tubuh pendek dan kaki yang
lebih pendek dari kucing biasa. Mereka memiliki penampilan unik yang lucu
dan aktif dalam bermain.'"),
    CatModel(
        urlImage:
            "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/bengal.png",
        name: "Bengal",
        desc:
            "'Bengal memiliki bulu yang berkilau dengan motif belang yang mirip
macan tutul. Mereka aktif, cerdas, dan suka bermain, sering kali memiliki
energi yang tinggi.'"),
    CatModel(
        urlImage:
            "https://raw.githubusercontent.com/Fahmisbas/acsl-
mcs/master/Bab%203%20-
%20RecyclerView%20%26%20OnClickListener/cats/britain%20shorthair.jpg",
        name: "Britain Shorthair",
        desc:
            "'British Shorthair memiliki penampilan gemuk dengan wajah yang

```

```
    bulat dan mata besar. Mereka cenderung tenang, santai, dan mudah diurus,
    membuat mereka menjadi kucing peliharaan yang populer.'''
  ];
```

Kita membuat data dummy di dalam suatu List. Data di dalam List dibuat dengan bentuk model dari CatModel dan isi dari List dengan memanggil class CatModel() yang mana CatModel() memiliki constructor untuk pengisian data.

Buka file home_screen.dart dan isikan code berikut :

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:mcs_bab_2/cat_data.dart';
import 'package:mcs_bab_2/cat_model.dart';
import 'package:mcs_bab_2/detail_screen.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Mcs Bab 2", style: TextStyle(color:
Colors.white)),
        backgroundColor: Colors.grey[900],
      ),
      body: ListView(
        children: [
          const SizedBox(height: 32,),

          Center(
            child: Text(
              "Jenis-jenis Kucing",
              style: GoogleFonts.roboto(
                fontSize: 16, fontWeight: FontWeight.w600,
              ),
            ),
          ),

          const SizedBox(height: 24,),

          ListView.builder(
            itemCount: cats.length,
            shrinkWrap: true,
            physics: const NeverScrollableScrollPhysics(),
            itemBuilder: (_, index) {
              CatModel cat = cats[index];
              return Container(
                margin: const EdgeInsets.symmetric(vertical: 12, horizontal:
18),
                child: GestureDetector(
```

```

        child: Row(
          children: [
            SizedBox(
              height: MediaQuery.of(context).size.width/4,
              width: MediaQuery.of(context).size.width/4,
              child: ClipRRect(
                borderRadius: BorderRadius.circular(18),
                child: Image.network(cat.urlImage, fit:
BoxFit.cover),
              ),
            ),
            const SizedBox(width: 14,),
            Text(cat.name),
          ],
        ),
        onTap: (){
          Navigator.push(
            context, MaterialPageRoute(builder: (context) {
              return DetailScreen(cat: cat);
            })),
        );
      },
    ),
  ),
),
],
),
);
}
}
}

```

ListView.builder() adalah widget yang dapat generate banyak widget di dalamnya secara otomatis sesuai ketersediaan data.

```

itemCount: cats.length,
shrinkWrap: true,
physics: const NeverScrollableScrollPhysics(),
itemBuilder: (_, index) {
  //...
}

```

ListView.builder() memiliki beberapa properti atau arguments yang harus diisi, itemCount untuk menentukan seberapa banyak data yang ada di dalam List, shrinkWrap untuk menentukan jika tempat dan ukuran dari widget ListView.builder() sudah ditentukan, physics: NeverScrollableScrollPhysics() agar widget ListView.builder() tidak dapat dilakukan scroll dan itemBuilder digunakan untuk membangun widget di dalam ListView.builder(). ListView.builder() di dalam aplikasi ini memiliki panjang sebanyak data yang ada pada variabel cats bertipe List diambil dari file cat_data.dart, bisa dilihat isi argument dari

itemCount: adalah cats.length. Deklarasi CatModel cat = cats[index] agar bisa mengambil data sesuai parameter index agar cukup dengan hanya menulis cat.<nama data>.

itemBuilder di dalam ListView.builder() memiliki return Container(). Container() memiliki child GestureDetector() agar pengguna dapat melakukan sebuah aksi pada widget tersebut. Aksi dari aksi yang dilakukan pengguna adalah perpindahan halaman untuk melihat detail data kucing. Halaman untuk melihat detail dari data kucing dibuat di class DetailScreen(). Di dalam class DetailScreen() terdapat Constructor untuk meminta data kucing untuk ditampilkan. Data yang dikirim dari class HomeScreen akan dikirim sesuai index. Data detail kucing diminta saat perpindahan halaman menuju DetailScreen() dengan mengisi argumen return DetailScreen(cat: cat);

Class DetailScreen() ditulis di dalam file bernama detail_screen.dart. Buka file detail_screen.dart dan tulis code berikut :

```
import 'package:flutter/material.dart';
import 'package:mcs_bab_2/cat_model.dart';

class DetailScreen extends StatefulWidget {
  CatModel cat;

  DetailScreen({super.key, required this.cat,});

  @override
  State<DetailScreen> createState() => _DetailScreenState();
}

class _DetailScreenState extends State<DetailScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: ListView(
        children: [
          Image.network(widget.cat.urlImage,),

          const SizedBox(height: 18,),

          Container(
            margin: const EdgeInsets.symmetric(horizontal: 18),
            child: Text(widget.cat.name),
          ),

          const SizedBox(height: 18,),

          Container(
            margin: const EdgeInsets.symmetric(horizontal: 18),
            child: Text(widget.cat.desc, textAlign: TextAlign.justify),
          ),
        ],
      ),
    ),
  ),
}
```

```
);  
}  
}
```

Halaman detail data kucing dibuat menggunakan StatefulWidget sehingga mempunyai 1 class extend dari State<DetailScreen> berisfta private mempunyai danam class _DetailScreenState(). Class _DetailScreenState() mengambil data kucing untuk ditampilkan dengan cara

```
Image.network(widget.cat.urlImage, ),  
  
const SizedBox(height: 18, ),  
  
Container(  
  margin: const EdgeInsets.symmetric(horizontal: 18),  
  child: Text(widget.cat.name),  
) ,  
  
const SizedBox(height: 18, ),  
  
Container(  
  margin: const EdgeInsets.symmetric(horizontal: 18),  
  child: Text(widget.cat.desc, textAlign: TextAlign.justify),
```

LAPORAN AWAL (LP)

1. Jelaskan apa itu dinamis UI!
2. Jelaskan apa itu data dummy!
3. Jelaskan apa itu ListView!
4. Seberapa penting dinamis UI pada pembuatan aplikasi?

LAPORAN AKHIR (LA)

1. Berikan kesimpulan pada Bab 2!