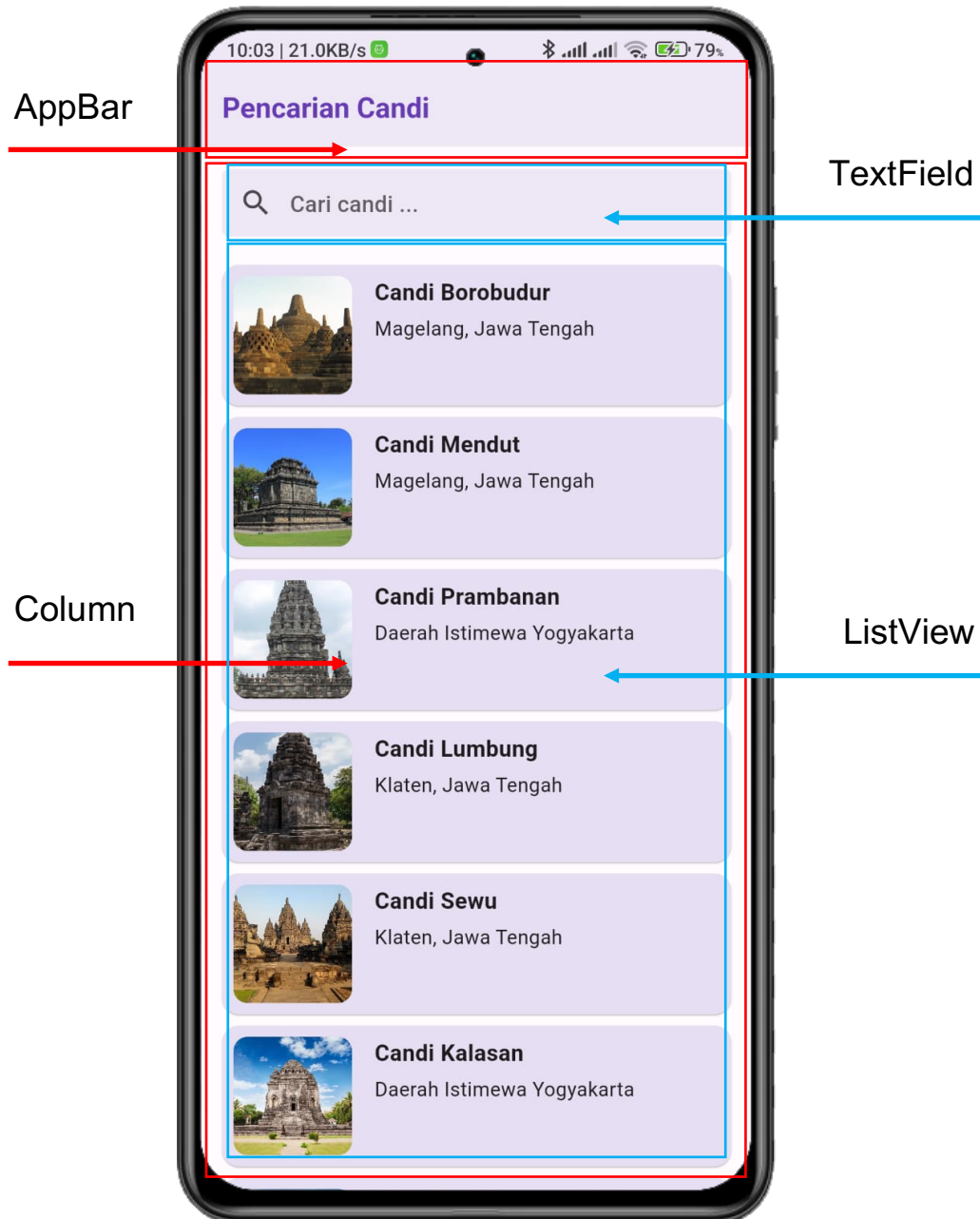


## ListView pada Implementasi SearchScreen



## Widget yang digunakan

- StatefulWidget
- Scaffold
- Container
- Padding
- Column
- TextField
- InputDecoration
- Icon
- OutlineInputBorder
- BoxDecoration
- Expanded
- ListView
- SizedBox
- Image
- ClipRRect
- Row

## Buat SearchScreen

Buat file baru bernama **search\_screen.dart** dalam direktori **lib/screens/**, lalu buat **StatefulWidget** dengan nama **SearchScreen**. Tentu saja, *widget* ini memerlukan paket **material**. Untuk itu, impor paket **material.dart**.

Untuk memudahkan kita melacak pekerjaan, buat **TODO** yang membagi tahapan pekerjaan.

1. TODO: 1. Deklarasikan variabel yang dibutuhkan.
2. TODO: 2. Buat appBar dengan judul Pencarian Candi
3. TODO: 3. Buat body berupa Column
4. TODO: 4. Buat TextField pencarian sebagai anak dari Column
5. TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column

Ubah **const Placeholder** yang menjadi kembalian *default* dengan **Scaffold**. Kemudian, letakkan **TODO: 2 – 5** di bawah hierarki **Scaffold**.

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

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

  @override
  State<SearchScreen> createState() => _SearchScreenState();
}

class _SearchScreenState extends State<SearchScreen> {
  // TODO: 1. Deklarasikan variabel yang dibutuhkan

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      // TODO: 2. Buat appBar dengan judul Pencarian Candi
      // TODO: 3. Buat body berupa Column
      // TODO: 4. Buat TextField pencarian sebagai anak dari Column
      // TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column
    );
  }
}
```

## Jalankan Aplikasi

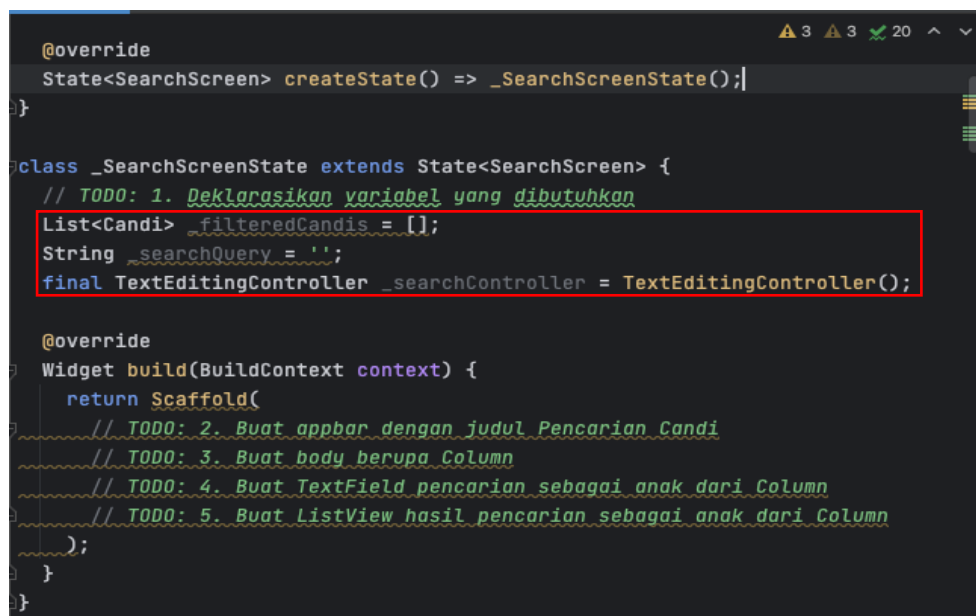
Pastikan aplikasi bisa berjalan sebelum mulai bekerja. Sebelum menjalankan aplikasi, ubahlah home pada **main.dart** menjadi **SearchScreen**. Perubahan ini membutuhkan untuk mengimpor **search\_screen.dart**. Setelah aplikasi di jalankan, pada emulator akan tampak layar putih kosong.

## Deklarasi Variabel

Kita mulai dengan **TODO: 1** yaitu mendeklarasikan variabel. Ada tiga variabel yang dibutuhkan.

1. `List<Candi> _filteredCandis = []`
2. `String _searchQuery = ''`
3. `final TextEditingController _searchController = TextEditingController()`

Variabel pertama membutuhkan objek Candi dari model data. Untuk itu, perlu mengimpor paket `'../models/candi.dart'`. Variabel ini akan digunakan untuk menampilkan data candi. Variabel kedua akan digunakan untuk menangkap nilai teks yang dimasukkan pengguna pada **TextField** sebagai teks yang dicari. Variabel ketiga adalah controller yang akan mengontrol masukan pengguna pada **TextField** pencarian. Kedua variabel terakhir akan digunakan kemudian ketika mengimplementasi fitur pencarian.



```
@override
State<SearchScreen> createState() => _SearchScreenState();
}

class _SearchScreenState extends State<SearchScreen> {
  // TODO: 1. Deklarasikan variabel yang dibutuhkan
  List<Candi> _filteredCandis = [];
  String _searchQuery = '';
  final TextEditingController _searchController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      // TODO: 2. Buat appBar dengan judul Pencarian Candi
      // TODO: 3. Buat body berupa Column
      // TODO: 4. Buat TextField pencarian sebagai anak dari Column
      // TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column
    );
  }
}
```

## AppBar dan Body

Implementasi layar ini menggunakan appBar. Buat appBar (**TODO: 2**) yang menjadi parameter dari **Scaffold** dengan menggunakan widget **AppBar** dengan title berupa **Text** dengan nilai 'Pencarian Candi.' Parameter lain dari **Scaffold** adalah body menggunakan **Column**.

```
class _SearchScreenState extends State<SearchScreen> {  
  // TODO: 1. Deklarasikan variabel yang dibutuhkan  
  List<Candi> _filteredCandis = [];  
  String _searchQuery = '';  
  final TextEditingController _searchController = TextEditingController();  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      // TODO: 2. Buat appBar dengan judul Pencarian Candi  
      appBar: AppBar(title: Text('Pencarian Candi'),),  
      // TODO: 3. Buat body berupa Column  
      body: Column(  
        children: [  
          // TODO: 4. Buat TextField pencarian sebagai anak dari Column  
          // TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column  
          1,  
        ],  
      ), // Column  
    ); // Scaffold  
  }  
}
```

## Bagian SearchScreen

**SearchScreen** terdiri dari dua bagian yaitu **TextField** untuk pencarian dan **ListView** untuk hasil pencarian. Untuk sementara, *widget* berupa **TextField** dan **ListView** dipasang pada layar dengan menunda implementasi pencarian. Kita sementara hanya akan menampilkan semua data candi pada **ListView**.

## TextField Pencarian

Buat **TextField** sebagai anak dari **Column** (**TODO: 4**). Beri parameter **TextField** berupa **autofocus: false** dan **decoration: InputDecoration**. Tambahkan parameter dari **InputDecoration** sebagai berikut.

- **hintText:** 'Cari candi ...'
- **prefixIcon:** **Icon(Icons.search)**
- **border:** **InputBorder.none**
- **focusedBorder:** **OutlineInputBorder** dengan warna **borderSide** **deepPurple**.
- **contentPadding:** **EdgeInsets.symmetric** dengan nilai **horizontal:** 16 dan **vertical:** 12

Untuk membuat tampilan **TextField** menjadi seperti pada contoh, bungkus **TextField** dengan **Container**. Beri parameter **Container** berupa decoration dengan **BoxDecoration**. Pada **BoxDecoration**, tambahkan parameter berikut.

- border radius circular: 5
- color: deepPurple[50]
- Bungkus Container dengan Padding
- EdgeInsets.all: 16

```
Padding(  
  padding: const EdgeInsets.all(16),  
  child: Container(  
    decoration: BoxDecoration(  
      borderRadius: BorderRadius.circular(5),  
      color: Colors.deepPurple[50],  
    ), // BoxDecoration  
    child: TextField(  
      autofocus: false,  
      decoration: InputDecoration(  
        hintText: 'Cari candi ...',  
        prefixIcon: Icon(Icons.search),  
        border: InputBorder.none,  
        focusedBorder: OutlineInputBorder(  
          borderSide: BorderSide(color: Colors.deepPurple),  
        ), // OutlineInputBorder  
        contentPadding:  
          EdgeInsets.symmetric(horizontal: 16, vertical: 12),  
      ), // InputDecoration  
    ), // TextField  
  ), // Container
```

## ListView Hasil Pencarian

Selanjutnya, kita akan melanjutkan pada pembuatan **ListView (TODO: 5)** yang nantinya akan menampilkan hasil pencarian. Buat **ListView** sebagai anak dari **Column**. Jika **ListView** langsung menjadi anak dari **Column**, itu akan menimbulkan kesalahan karena dianggap menempatkan *scrollable widget* di dalam *scrollable widget*. Untuk mengatasi hal ini, bungkus **ListView** dengan **Expanded**.

Beri parameter **ListView** berupa `itemCount` dan `itemBuilder`. Setelah itu, Buat variabel lokal bernama `candi` yang berisi `_filteredCandis[index]`. Buat kembalian dari **ListView builder** berupa **Card**. **Card** ini yang akan menjadi tempat *list item* berupa **Image** dan dua **Text** yang berisi gambar candi, nama candi, dan lokasi candi.

```

appBar: AppBar(
  title: Text('Pencarian Candi'),
), // AppBar
// TODO: 3. Buat body berupa Column
body: Column(
  children: [
    // TODO: 4. Buat TextField pencarian sebagai anak dari Column
    Padding(...), // Padding
    // TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column
    ListView.builder(
      itemCount: _filteredCandis.length,
      itemBuilder: (context, index) {
        final candi = _filteredCandis[index];
        return Card();
      },
    ), // ListView.builder
  ],
), // Column
); // Scaffold
}

```

Sebagai parameter dari Card, tambahkan margin berupa `EdgeInsets.symmetric` dengan nilai horizontal: 16 dan vertical: 4. Tambahkan juga anak dari **Card** berupa **Row**. Beri parameter `crossAxisAlignment` (**`CrossAxisAlignment.start`**) pada **Row** untuk mengatur anak-anaknya menjadi rata kiri. **Row** mempunyai dua anak yaitu **Image** dan **Column**. Anak pertama dari **Row** adalah **Image asset** dengan nilai `candi.imageAsset`. Beri parameter `fit` dengan nilai **`BoxFit.cover`**.

Agar tampilan gambar jadi lebih menarik, bungkus **Image** dengan **`ClipRRect`** untuk membuat keempat sudutnya membulat. Gunakan parameter `borderRadius` dengan nilai **`BorderRadius.circular`** dan mengatur radius dengan nilai 10.

Untuk bisa mengatur tinggi dan lebar gambar dengan tepat, bungkus **`ClipRRect`** dengan **`Container`**. Tentukan nilai parameter `height` dan `width` sebesar 100. Tentukan juga nilai `padding` di semua sisi sebesar 8.

```
// TODO: 5. Buat ListView hasil pencarian sebagai a
Expanded(
  child: ListView.builder(
    itemCount: _filteredCandis.length,
    itemBuilder: (context, index) {
      final candi = _filteredCandis[index];
      // TODO: 8. Implementasi GestureDetector dan Hero animation
      return Card(
        margin: EdgeInsets.symmetric(horizontal: 16, vertical: 4),
        child: Row(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
              padding: EdgeInsets.all(8),
              width: 100,
              height: 100,
              child: ClipRRect(
                borderRadius: BorderRadius.circular(10),
                child: Image.asset(
                  candi.imageAsset,
                  fit: BoxFit.cover,
                ), // Image.asset
              ), // ClipRRect
            ), // Container
          ],
        ),
      );
    },
  ),
);
```

Selanjutnya, menambahkan **Column** sebagai anak kedua dari **Row**. Beri parameter `crossAxisAlignment` pada **Column** dengan nilai **CrossAxisAlignment.start**.

**Column** memiliki tiga anak yaitu **Text**, **SizedBox**, dan **Text**. **Text** pertama diisi dengan nilai **candi.name**. Atur ukuran font sebesar 16 dan ketebalannya bold. Di bawah **Text** pertama, tempatkan **SizedBox** yang diberi nilai `height: 4`. Kemudian, **Text** kedua diisi dengan **candi.location**.

Bungkus **Column** dengan **Padding** untuk memberi jarak dengan *widget* induk dan **Image**. Gunakan **EdgeInsets.all** dengan nilai 8.

Untuk mengetes data candi tampil pada **ListView**, ubah nilai variabel **List<Candi> \_filteredCandis = candiList;**

```

      child: Image.asset(
        candi.imageAsset,
        fit: BoxFit.cover,
      ), // Image.asset
    ), // ClipRRect
  ), // Container
  Padding(
    padding: const EdgeInsets.all(8.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(candi.name, style: TextStyle(
          fontSize: 16, fontWeight: FontWeight.bold,
        )), // TextStyle, Text
        SizedBox(height: 4),
        Text(candi.location),
      ],
    ), // Column
  ), // Padding
],
), // Row

```

## Implementasi Lanjutan

- Fitur pencarian (**TODO: 6**)
- Pengosongan nilai input (**TODO: 7**)
- **GestureDetector** dan **Hero** animation (**TODO: 8**)

```

    autofocus: false,
    // TODO: 6. Implementasi fitur pencarian
    decoration: InputDecoration(
      hintText: 'Cari candi ...',
      prefixIcon: Icon(Icons.search),
      // TODO: 7. Implementasi pengosongan input
      border: InputBorder.none,
      focusedBorder: OutlineInputBorder(...), // OutlineInputBorder
      contentPadding:
        EdgeInsets.symmetric(horizontal: 16, vertical: 12),
    ), // InputDecoration
  ), // TextField
), // Container
), // Padding
// TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column
Expanded(
  child: ListView.builder(
    itemCount: _filteredCandis.length,
    itemBuilder: (context, index) {
      final candi = _filteredCandis[index];
      // TODO: 8. Implementasi GestureDetector dan Hero animation
    },
  ),
),

```



## Kode Lengkap

```
import 'package:flutter/material.dart';
import 'package:wisata_candi/data/candi_data.dart';

import '../models/candi.dart';

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

  @override
  State<SearchScreen> createState() => _SearchScreenState();
}

class _SearchScreenState extends State<SearchScreen> {
  // TODO: 1. Deklarasikan variabel yang dibutuhkan
  final List<Candi> _filteredCandis = candiList;
  final String _searchQuery = '';
  final TextEditingController _searchController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      // TODO: 2. Buat appBar dengan judul Pencarian Candi
      appBar: AppBar(
        title: const Text('Pencarian Candi'),
      ),
      // TODO: 3. Buat body berupa Column
      body: Column(
        children: [
          // TODO: 4. Buat TextField pencarian sebagai anak dari Column
          Padding(
            padding: const EdgeInsets.all(16),
            child: Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(5),
                color: Colors.deepPurple[50],
              ),
              child: const TextField(
                autofocus: false,
                // TODO: 6. Implementasi fitur pencarian
                decoration: InputDecoration(
                  hintText: 'Cari candi ...',
                  prefixIcon: Icon(Icons.search),
                  // TODO: 7. Implementasi pengosongan input
                  border: InputBorder.none,
                  focusedBorder: OutlineInputBorder(
                    borderSide: BorderSide(color: Colors.deepPurple),
                  ),
                ),
                contentPadding:
                  EdgeInsets.symmetric(horizontal: 16, vertical: 12),
              ),
            ),
          ),
          // TODO: 5. Buat ListView hasil pencarian sebagai anak dari Column

```

```

Expanded(
  child: ListView.builder(
    itemCount: _filteredCandis.length,
    itemBuilder: (_, index) {
      final candi = _filteredCandis[index];
      // TODO: 8. Implementasi GestureDetector dan Hero animation
      return Card(
        margin: const EdgeInsets.symmetric(horizontal: 16,
vertical: 4),
        child: Row(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
              padding: const EdgeInsets.all(8),
              width: 100,
              height: 100,
              child: ClipRRect(
                borderRadius: BorderRadius.circular(10),
                child: Image.asset(
                  candi.imageAsset,
                  fit: BoxFit.cover,
                ),
              ),
            ),
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(candi.name, style: const TextStyle(
                    fontSize: 16, fontWeight: FontWeight.bold,
                  )),
                  const SizedBox(height: 4),
                  Text(candi.location),
                ],
              ),
            ),
          ],
        ),
      );
    },
  ),
  const SizedBox(height: 16)
),
);
}
}

```