

## PANDUAN TUGAS 4

### Membuat Tampilan Home

#### Capaian

1. Mahasiswa mampu menggunakan dan menampilkan asset lokal berformat svg dalam proyek flutter dengan memanfaatkan package flutter\_svg.
2. Mahasiswa dapat menampilkan Text widget dengan style yang memanfaatkan package google\_fonts.
3. Mahasiswa dapat menggunakan Icon widget.
4. Mahasiswa mampu menggunakan Image widget untuk menampilkan asset gambar serta memanfaatkan package cached\_network\_image untuk menampilkan gambar dari internet.
5. Mahasiswa dapat melakukan manipulasi data string berformat dengan memanfaatkan extension function.

#### Spesifikasi Hardware dan Software

Memiliki komponen perangkat keras dan perangkat lunak yang benar sangat penting untuk memastikan keberhasilan pelaksanaan tugas yang diuraikan dalam panduan ini. Konfigurasi perangkat keras dan perangkat lunak yang diperlukan untuk menyelesaikan tugas panduan ini adalah sebagai berikut:

##### A. Spesifikasi Minimum Hardware

1. Minimum RAM 4 GB, disarankan RAM 8 GB
2. Minimum 15 GB ruang disk yang tersedia (2 GB untuk Flutter SDK, 8 GB untuk Android Studio, 4 GB untuk AVD, dan 1 GB untuk proyek)
3. Resolusi layar minimum 1280 x 800
4. Emulator Mobile (Android/IOS)

##### B. Software

1. Flutter SDK (versi 3.13.0 atau yang lebih baru) dan Dart (versi 3.1.5 atau yang lebih baru)
2. Android Studio
3. Visual Studio Code

#### Sumber Daya

1. [Test File Tugas 2](#)

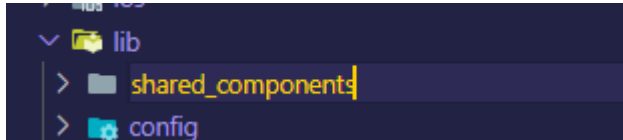
#### Deskripsi Tugas

Mahasiswa akan menuliskan kode pada proyek Flutter media\_player untuk mengimplementasikan komponen UI dasar multimedia. Mahasiswa perlu memahami cara pembuatan komponen yang dapat menampilkan gambar baik dari lokal asset maupun internet, teks, icon, serta berbagai dekorasi yang dapat ditambahkan. Dalam panduan ini mahasiswa akan membuat komponen CustomAppBar, CoverMusicCard, CoverVideoCard, TitleSection,

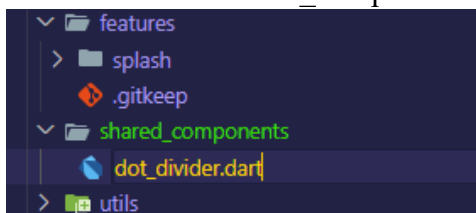
DotDivider, dan halaman Home. Selain itu, mahasiswa juga akan menampilkan hasil manipulasi teks berformat dari proses fetching data dengan memanfaatkan ekstension function.

### Langkah Praktikum 1 (Membuat Komponen CustomAppBar, TitleSection, dan DotDivider)

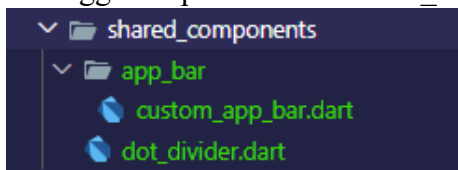
1. Buka proyek flutter media\_player pada Visual Studio Code anda.
2. Buka/Pilih folder lib. Kemudian buat folder baru dengan nama shared\_components.



3. Di dalam folder shared\_components tersebut, buat file baru bernama dot\_divider.dart.



4. Masih di dalam folder shared\_components, buat folder baru dengan nama app\_bar lalu buat file baru dengan nama custom\_app\_bar.dart di dalam folder app\_bar tersebut. Sehingga tampilan folder shared\_components akan terlihat seperti berikut.

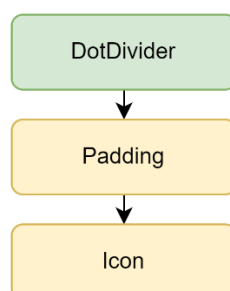


5. Buat sebuah stateless widget dengan nama DotDivider di dalam file dot\_divider.dart.

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

class DotDivider extends StatelessWidget {
  const DotDivider({super.key});
  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

6. Ubah widget kembalian dari DotDivider menjadi seperti berikut.

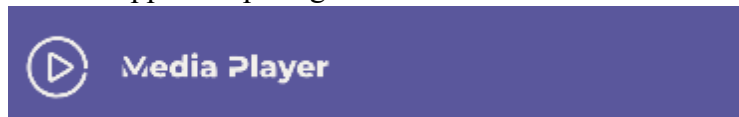


```
Widget build(BuildContext context) {
  return const Padding(
    padding: padding,
    child: Icon(icon),
  );
}
```

7. Ubah properti dari setiap widget dalam DotDivider tersebut sesuai dengan spesifikasi berikut.

Widget	Properti	Nilai
Padding	padding	EdgeInsets.symmetric(horizontal: 6)
	child	Icon()
Icon	icon	Icons.circle
	size	6
	color	MainColor.whiteF2F0EB

8. Setelah selesai membuat komponen DotDivider. Anda akan membuat komponen CustomAppBar seperti gambar di bawah ini.



9. Buka file **custom\_app\_bar.dart**. Buat stateless widget dengan nama **CustomAppBar** dalam file tersebut.

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

class CustomAppBar extends StatelessWidget {
  const CustomAppBar({super.key});

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

10. Tambahkan kode `implements PreferredSizeWidget` pada kelas CustomAppBar. Kode tersebut digunakan untuk mengimplementasikan antarmuka (interface) PreferredSizeWidget dengan memaksa pendefinisian ulang fungsi (override).

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

class CustomAppBar extends StatelessWidget implements PreferredSizeWidget {
  const CustomAppBar({super.key});

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}

package:media_player/shared_components/app_bar/custom_app_bar.dart
```

11. Setelah menambahkan implementasi PreferredSizeWidget pada kelas CustomAppBar, anda perlu menambahkan kode yang meng-override getter preferredSize di dalam kelas CustomAppBar.

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

class CustomAppBar extends StatelessWidget implements PreferredSizeWidget {
  const CustomAppBar({super.key});

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }

  @override
  Size get preferredSize => const Size.fromHeight(56);
}
```

12. Tambahkan properti bertipe nullable Widget leading dan title pada CustomAppBar.

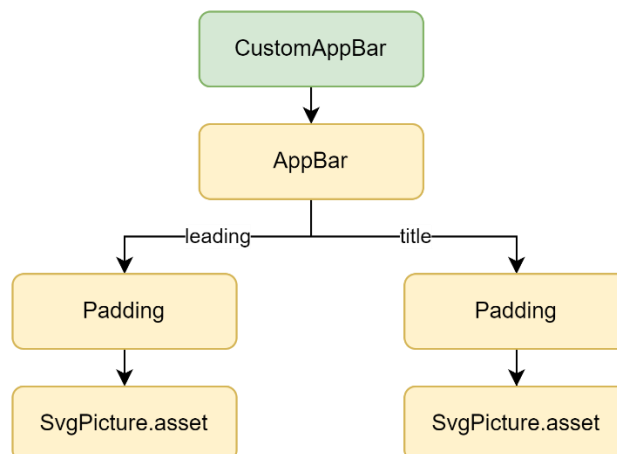
```
final Widget? leading;
final Widget? title;
```

13. Tambahkan properti tersebut ke dalam konstruktor CustomAppBar sebagai parameternya.

```
class CustomAppBar extends StatelessWidget implements PreferredSizeWidget {
  const CustomAppBar({
    super.key,
    this.leading,
    this.title,
  });

  final Widget? leading;
  final Widget? title;
}
```

14. Ubah widget kembalian dari CustomAppBar menjadi seperti berikut.



```
Widget build(BuildContext context) {
  return const AppBar(
    leading: Padding(
      child: SvgPicture.asset(),
```

```

    ),
    title: Padding(
      child: SvgPicture.asset(),
    ),
  );
}

```

15. Ubah properti dari setiap widget dalam CustomAppBar tersebut sesuai dengan spesifikasi berikut.

Widget	Properti	Nilai
AppBar	backgroundColor	MainColor.purple5A579C
	leading	Padding()
	titleSpacing	0
	title	Padding()
[leading] Padding	padding	const EdgeInsets.all(8)
	child	SvgPicture.asset()
[child of leading widget] SvgPicture.asset	assetName	AssetsConsts.logo
	colorFilter	const ColorFilter.mode(           MainColor.whiteF2F0EB,           BlendMode.srcIn,         )
[title] Padding	padding	EdgeInsets.symmetric(           vertical: 8,         )
	child	SvgPicture.asset()
[child of title widget] SvgPicture.asset	assetName	AssetsConsts.logoName
	width	120
	colorFilter	const ColorFilter.mode(           MainColor.whiteF2F0EB,           BlendMode.srcIn,         )

16. Pada [leading] Padding widget, tambahkan operator null-aware ?? pada argumen child-nya agar bisa menampilkan widget yang nantinya diberikan melalui **leading** argumen saat pembuatan/pemanggilan CustomAppBar. Jika **leading** bernilai null, maka tampilkan SvgPicture.asset widget yang sudah anda buat sebelumnya.

17. Tambahkan juga operator null-aware ?? pada argumen child dari [title] Padding agar bisa menampilkan widget yang nantinya diberikan melalui **title** argumen saat pembuatan/pemanggilan CustomAppBar. Jika **title** bernilai null, maka tampilkan SvgPicture.asset widget yang sudah anda buat sebelumnya.

18. Berikut adalah kode akhir dari file custom\_app\_bar.dart

```

class CustomAppBar extends StatelessWidget implements PreferredSizeWidget {
  const CustomAppBar({
    super.key,
    this.leading,
    this.title,
  });

  final Widget? leading;
  final Widget? title;

  @override
  Widget build(BuildContext context) {
    return AppBar(
      backgroundColor: □MainColor.purple5A579C,
      leading: Padding(
        padding: const EdgeInsets.all(8),
        child: leading ??
          SvgPicture.asset(
            AssetsConsts.logo,
            colorFilter: const ColorFilter.mode(
              ■MainColor.whiteF2F0EB,
              BlendMode.srcIn,
            ), // ColorFilter.mode
          ), // SvgPicture.asset
      ), // Padding
      titleSpacing: 0,
      title: Padding(
        padding: const EdgeInsets.symmetric(
          vertical: 8,
        ), // EdgeInsets.symmetric
        child: title ??
          SvgPicture.asset(
            AssetsConsts.logoName,
            width: 120,
            colorFilter: const ColorFilter.mode(
              ■MainColor.whiteF2F0EB,
              BlendMode.srcIn,
            ), // ColorFilter.mode
          ), // SvgPicture.asset
      ), // Padding
    ); // AppBar
  }

  @override
  Size get preferredSize => const Size.fromHeight(56);
}

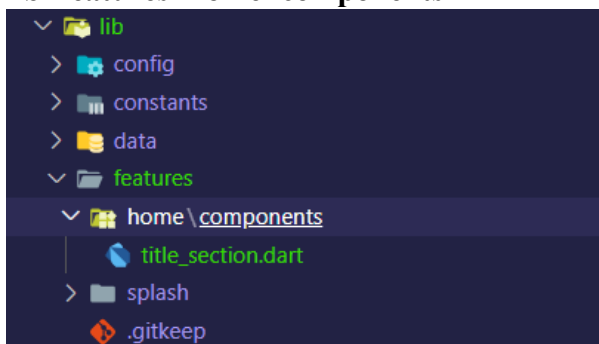
```

19. Selanjutnya anda akan membuat komponen TitleSection yang akan terlihat seperti gambar berikut.



**Music Collections**

20. Buat file baru dengan nama **title\_section.dart** di dalam folder **lib>features>home>components**.



21. Buat sebuah stateless widget dengan nama **TitleSection** di dalam file `title_section.dart`.

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

class TitleSection extends StatelessWidget {
  const TitleSection({super.key});

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

22. Tambahkan properti String `title` pada `TitleSection`. Jangan lupa untuk menambahkan properti tersebut ke dalam konstruktornya juga sebagai parameter.

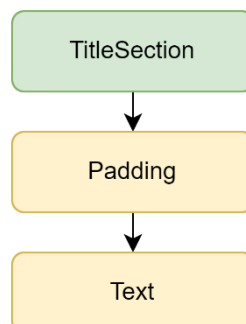
```
import 'package:flutter/material.dart';

class TitleSection extends StatelessWidget {
  const TitleSection({
    super.key,
    required this.title,
  });

  final String title;

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

23. Ubah widget kembalian dari `TitleSection` menjadi seperti berikut.



```
@override
Widget build(BuildContext context) {
  return Padding(
    child: Text(),
  );
}
```

24. Ubah properti dari setiap widget dalam TitleSection tersebut sesuai dengan spesifikasi berikut.

Widget	Properti	Nilai
Padding	padding	const EdgeInsets.symmetric(horizontal: 15, vertical: 10)
	child	Text()
Text	data	title
	style	MainTextStyle.poppinsW600.copyWith( fontSize: 20, color: MainColor.whiteF2F0EB, )

25. Berikut adalah kode akhir dari komponen title\_section.dart

```
import 'package:flutter/material.dart';
import 'package:media_player/config/themes/main_color.dart';
import 'package:media_player/config/themes/main_text_style.dart';

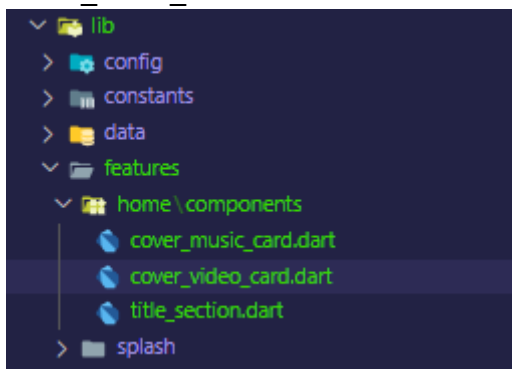
class TitleSection extends StatelessWidget {
  const TitleSection({
    super.key,
    required this.title,
  });

  final String title;

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15, vertical: 10),
      child: Text(
        title,
        style: MainTextStyle.poppinsW600.copyWith(
          fontSize: 20,
          color: MainColor.whiteF2F0EB,
        ),
      ), // Text
    ); // Padding
  }
}
```

## Langkah Praktikum 2 (Membuat Komponen CoverMusicCard)

1. Pada folder yang sama dengan peletakkan file title\_section.dart, yaitu lib>features>home>components, Buat file baru dengan nama **cover\_music\_card.dart** dan **cover\_video\_card.dart**.





2. Berikut adalah tampilan dari komponen CoverMusicCard yang akan anda buat.



3. Buka file cover\_music\_card.dart. Kemudian buat sebuah stateless widget dengan nama **CoverMusicCard**.

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

class CoverMusicCard extends StatelessWidget {
  const CoverMusicCard({super.key});

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

4. Tambahkan properti bertipe Music dengan nama music ke dalam CoverMusicCard. Jangan lupa untuk menambahkannya juga ke dalam konstruktor.

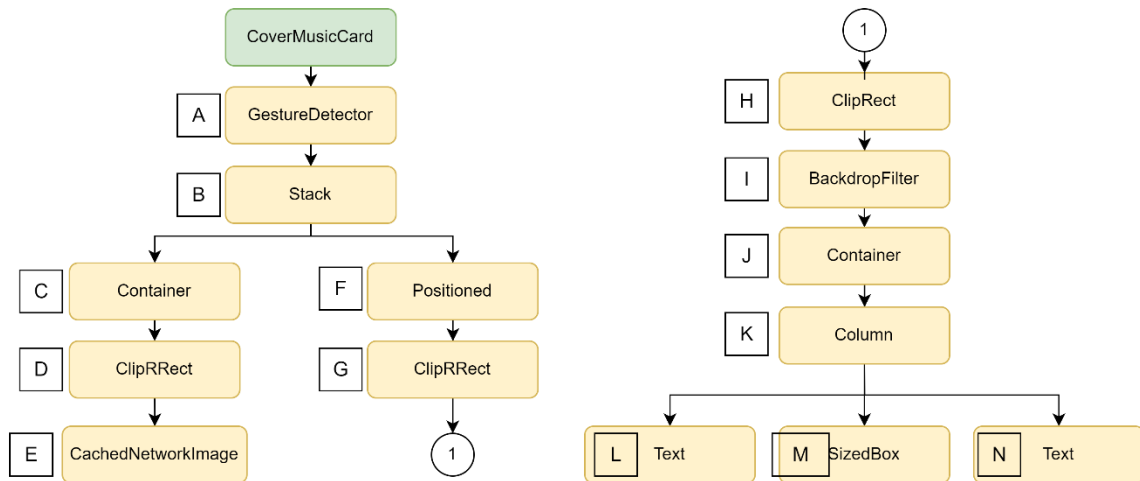
```
import 'package:flutter/material.dart';
import 'package:media_player/data/music_model.dart';

class CoverMusicCard extends StatelessWidget {
  const CoverMusicCard({
    super.key,
    required this.music,
  });

  final Music music;

  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

5. Ubah widget kembalian dari CoverMusicCard menjadi seperti berikut.



```
Widget build(BuildContext context) {
  return GestureDetector(
    child: Stack(
      children: [
        Container(
          child: ClipRRect(
            child: CachedNetworkImage(),
          ),
        ),
        Positioned(
          child: ClipRRect(
            child: ClipRect(
              child: BackdropFilter(
                child: Container(
                  child: Column(
                    children: [
                      Text(),
                      SizedBox(),
                      Text(),
                    ],
                  ),
                ),
            ),
          ),
        ),
      ],
    ),
  );
}
```

6. Tambahkan properti dari setiap widget dalam CoverMusicCard tersebut sesuai dengan spesifikasi berikut.

[KODE] Widget	Properti	Nilai
[A] GestureDetector	onTap	() {}
[B] Stack	alignment	Alignment.bottomCenter
[C] Container	width	190
	height	200
	decoration	ShapeDecoration(

		<pre>image: DecorationImage(   image: AssetImage(     music.coverPath!,   ),   fit: BoxFit.cover, ), shape: RoundedRectangleBorder(   borderRadius: BorderRadius.circular(36), ), )</pre>
[D] ClipRRect	borderRadius	BorderRadius.circular(36)
[E] CachedNetworkImage	imageUrl	music.coverPath!
	progressIndicatorBuilder	<pre>(context, url, progress) =&gt;   const Center(     child: CircularProgressIndicator(       color: MainColor.purple5A579C,     ),   )</pre>
	fit	Boxfit.cover
[F] Positioned	bottom	-1
[G] ClipRRect	borderRadius	<pre>const BorderRadius.only(   bottomLeft: Radius.circular(36),   bottomRight: Radius.circular(36), )</pre>
[I] BackdropFilter	filter	ImageFilter.blur(sigmaX: 5, sigmaY: 5)
[J] Container	alignment	Alignment.centerLeft
	width	190
	height	60
	padding	<pre>const EdgeInsets.symmetric(   horizontal: 24, vertical: 8, )</pre>
	decoration	<pre>const BoxDecoration(   gradient: LinearGradient(     begin: Alignment(0.00, -1.00),     end: Alignment(0, 1),     colors: [       MainColor.black120911,       MainColor.black0D0D0D,     ],   ), )</pre>
[K] Column	crossAxisAlignment	CrossAxisAlignment.start
[L] Text	data	music.title!
	maxLines	1
	overflow	TextOverflow.ellipsis

	style	MainTextStyle.poppinsW600.copyWith( fontSize: 14, color: MainColor.whiteF2F0EB, )
[M] SizedBox	height	3
[N] Text	data	music.artist!
	maxLines	1
	overflow	TextOverflow.ellipsis
	style	MainTextStyle.poppinsW400.copyWith( fontSize: 12, letterSpacing: 0.4, color: MainColor.whiteF2F0EB, )

7. Nilai properti decoration dari Container [C] berfungsi untuk menampilkan gambar cover musik yang bersumber dari asset lokal.

```
Container(
  width: 190,
  height: 200,
  decoration: ShapeDecoration(
    image: DecorationImage(
      image: AssetImage(
        music.coverPath!,
      ), // AssetImage
      fit: BoxFit.cover,
    ), // DecorationImage
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(36),
    ), // RoundedRectangleBorder
  ), // ShapeDecoration
)
```

8. Sedangkan widget CachedNetworkImage [E] berfungsi untuk menampilkan gambar cover musik yang bersumber dari internet/network.

```
child: ClipRRect(
  borderRadius: BorderRadius.circular(36),
  child: CachedNetworkImage(
    imageUrl: music.coverPath!,
    progressIndicatorBuilder: (context, url, progress) =>
      const Center(
        child: CircularProgressIndicator(
          color: MainColor.purple5A579C,
        ), // CircularProgressIndicator
      ), // Center
    fit: BoxFit.cover,
  ), // CachedNetworkImage
), // ClipRRect
```

9. Agar CoverMusicCard bisa mengatur widget mana yang akan ditampilkan berdasarkan sumber gambar dari cover musik yang diberikan (internet atau asset lokal), anda perlu

menambahkan ternary operator. Sehingga properti decoration dan child dari widget Container [C] akan terlihat seperti berikut.

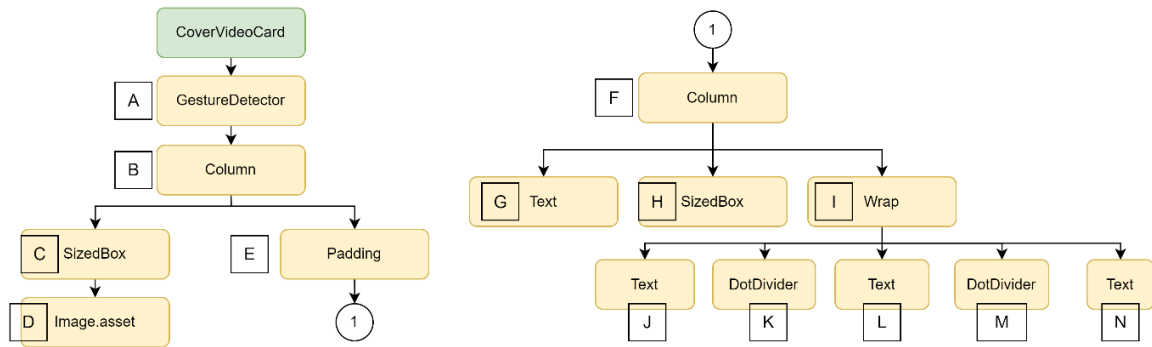
```
Container(  
  width: 190,  
  height: 200,  
  decoration: music.sourceType == "local"  
    ? ShapeDecoration( // ShapeDecoration ...  
      : null,  
    child: music.sourceType != "local"  
      ? ClipRRect( // ClipRRect ...  
        : null,  
      ), // Container  
) Positioned( // Positioned ...
```

### Langkah Praktikum 3 (Membuat Komponen CoverVideoCard)

1. Pada praktikum 3 ini, anda akan membuat komponen CoverVideoCard seperti gambar di bawah ini.



2. Buka file cover\_video\_card.dart. Kemudian buat sebuah stateless widget dengan nama **CoverVideoCard**.
3. Tambahkan properti bertipe Video dengan nama video ke dalam CoverVideoCard. Jangan lupa untuk menambahkannya juga ke dalam konstruktor.
4. Ubah widget kembalian dari CoverVideoCard menjadi seperti berikut.



5. Tambahkan properti dari setiap widget dalam CoverVideoCard tersebut sesuai dengan spesifikasi berikut.

[KODE] Widget	Properti	Nilai
[A] GestureDetector	onTap	() {}
[B] Column	key	const Key('column_card_wrapper')
	crossAxisAlignment	CrossAxisAlignment.start
[C] SizedBox	width	double.infinity
	height	MediaQuery.sizeOf(context).width * 9 / 16
[D] Image.asset	name	video.coverPath!
	fit	BoxFit.cover
[E] Padding	padding	const EdgeInsets.symmetric( vertical: 8, horizontal: 24, )
[F] Column	crossAxisAlignment	CrossAxisAlignment.start
[G] Text	data	video.title!
	maxLines	2
	overflow	TextOverflow.ellipsis
	style	MainTextStyle.poppinsW700.copyWith( fontSize: 15, color: MainColor.whiteF2F0EB, )
[H] SizedBox	height	4
[I] Wrap	crossAxisAlignment	WrapCrossAlignment.center
	runSpacing	12
[J] Text	data	video.creator!
	overflow	TextOverflow.ellipsis
	style	MainTextStyle.poppinsW400.copyWith( fontSize: 12, color: MainColor.whiteF2F0EB, )
[L] Text	data	"\${video.viewsCount!.formatViewsCount()} x views"
	overflow	TextOverflow.ellipsis
	style	MainTextStyle.poppinsW400.copyWith( fontSize: 12, color: MainColor.whiteF2F0EB, )

		)
[N] Text	data	video.releaseDate!.toLocalTime()
	overflow	TextOverflow.ellipsis
	style	MainTextStyle.poppinsW400.copyWith( fontSize: 12, color: MainColor.whiteF2F0EB, )

6. Nilai dari properti child widget SizedBox [C] digunakan untuk menampilkan menampilkan gambar cover video yang bersumber dari asset lokal.

```

SizedBox(
  width: double.infinity,
  height: MediaQuery.sizeOf(context).width * 9 / 16,
  child: Image.asset(
    video.coverPath!,
    fit: BoxFit.cover,
  ), // Image.asset
), // SizedBox

```

7. Agar CoverVideoCard bisa menampilkan gambar dari internet juga, anda perlu membuat widget baru untuk nantinya dijadikan child SizedBox [C] dengan memanfaatkan ternary operator. Berikut adalah spesifikasi widget untuk menampilkan gambar dari internet.

Widget	Properti	Nilai
ClipRRect	child	CachedNetworkImage()
CachedNetworkImage	imageUrl	video.coverPath!
	fit	BoxFit.cover

8. Agar CoverVideoCard bisa mengatur widget mana yang akan ditampilkan berdasarkan sumber gambar dari cover video yang diberikan (internet atau asset lokal), anda perlu menambahkan ternary operator dengan kondisi `video.sourceType == "local"`. Sehingga widget SizedBox [C] akan terlihat seperti berikut.

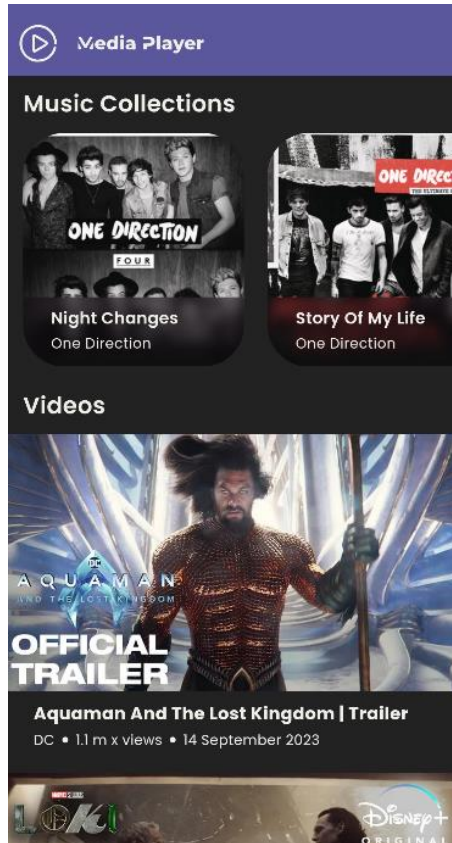
```

SizedBox(
  width: double.infinity,
  height: MediaQuery.sizeOf(context).width * 9 / 16,
  child: video.sourceType == "local"
    ? Image.asset(
      video.coverPath!,
      fit: BoxFit.cover,
    ) // Image.asset
    : ClipRRect(
      child: CachedNetworkImage(
        imageUrl: video.coverPath!,
        fit: BoxFit.cover,
      ), // CachedNetworkImage
    ), // ClipRRect
), // SizedBox

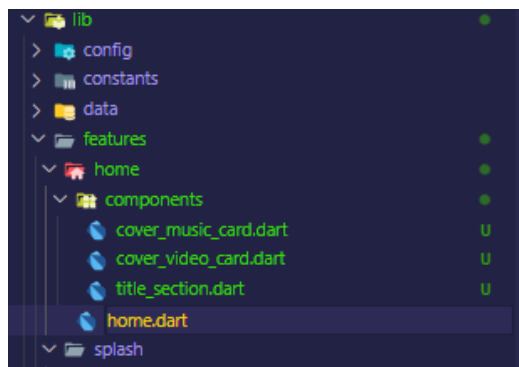
```

## Langkah Praktikum 4 (Membuat Tampilan Home)

1. Pada pratikum 4 ini, anda akan membuat halaman Home yang tersusun dari komponen-komponen yang telah anda buat pada praktikum sebelumnya seperti CustomAppBar, TitleSection, CoverMusicCard, dan CoverVideoCard. Berikut adalah tampilan Home yang akan anda buat.



2. Buat file baru di dalam folder lib>features>home dengan nama **home.dart**.



3. Pada file tersebut, buat sebuah stateful widget dengan nama **Home**.



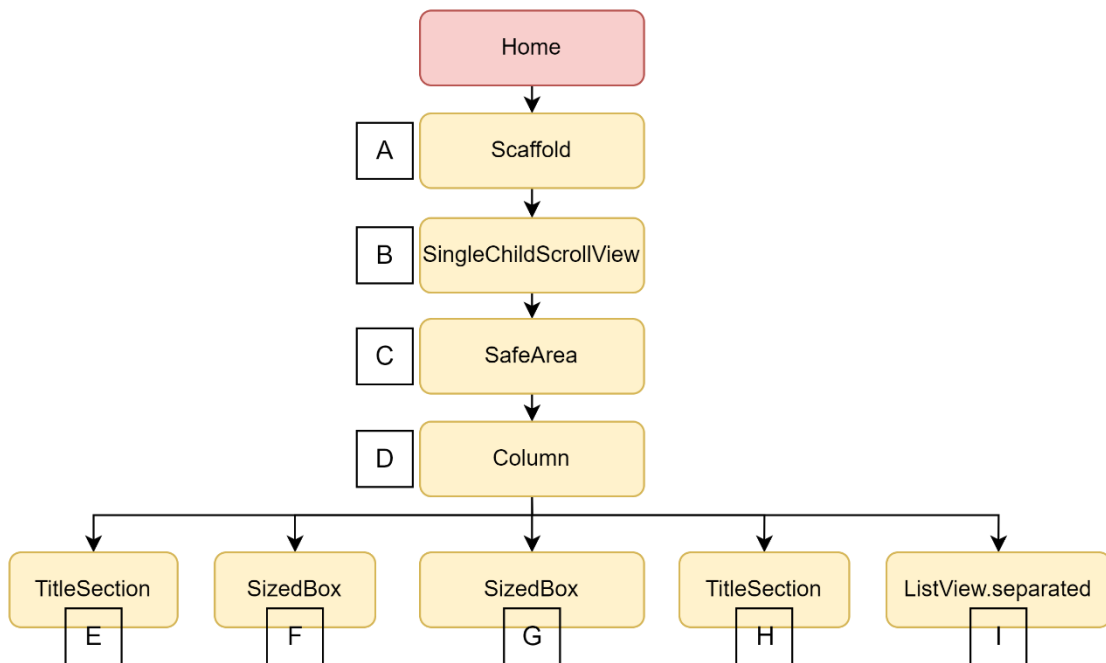
```
import 'package:flutter/material.dart';

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

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {
    return const Placeholder();
  }
}
```

4. Kemudian ubah widget kembalian dari Home menjadi seperti berikut.



```
class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: SafeArea(
          child: Column(
            children: [
              TitleSection(),
              SizedBox(),
              SizedBox(),
              TitleSection(),
              ListView.separated(),
            ],
          ), // Column
        ), // SafeArea
      ), // SingleChildScrollView
    ); // Scaffold
  }
}
```

\*Jika dalam menulis kode flutter anda mengalami error, anda dapat menekan tombol **CTRL** + **.**. Shortcut tersebut berguna untuk melakukan Quick Fix / Quick Action.

```

Rec Undefined name 'MainColor'.
us: Try correcting the name to one that is defined, or defining the
hed name. dart(undefined_identifier)
: C
Ind Type: Invalid type
Ci View Problem (Alt+F8) Quick Fix... (Ctrl+.)
or: MainColor.pur 1e5A579C, Undefined name 'MainColor'. Try corre
e: progress, progress.
  
```

5. Untuk melengkapi kode sebelumnya, anda perlu menambahkan properti setiap widget yang ada pada halaman Home agar sesuai dengan spesifikasi berikut.

[KODE] Widget	Properti	Nilai
[A] Scaffold	backgroundColor	MainColor.black222222
	appBar	const CustomAppBar()
[D] Column	mainAxisAlignment	MainAxisAlignment.start
	crossAxisAlignment	CrossAxisAlignment.start
[E] TitleSection	title	'Music Collections'
[F] SizedBox	height	200
[G] SizedBox	height	10
[H] TitleSection	title	'Videos'
[I] ListView.separated	key	const Key('video_list_view')
	physics	const NeverScrollableScrollPhysics()
	shrinkWrap	true
	separatorBuilder	(context, index) => const SizedBox(           height: 12,         )

6. Perhatikan kode pada widget ListView.separated [I]. Anda akan mendapatkan error seperti gambar berikut.

```

The named parameter 'itemBuilder' is required, but there's no
corresponding argument.
Try adding the required argument. dart(missing_required_argument)

The named parameter 'itemCount' is required, but there's no corresponding
argument.
Try adding the required argument. dart(missing_required_argument)

(new) ListView ListView.separated({
  Key? key,
  Axis scrollDirection = Axis.vertical,
  bool reverse = false,
  ListView.separated(
    The named parameter 'itemBuilder' is required, but there
    key: const Key('video_list_view'),
    physics: const NeverScrollableScrollPhysics(),
    shrinkWrap: true,
    separatorBuilder: (context, index) => const SizedBox(
      height: 12,
    ), // SizedBox
  ), // ListView.separated
)
  
```

Maksud dari error di atas adalah anda perlu menambahkan properti/argumen **itemBuilder** dan **itemCount** pada `ListView.separated`.

7. Untuk melengkapi kode yang diperlukan `ListView.separated`, anda perlu menambahkan state baru untuk menyimpan data musik dan video yang akan ditampilkan. Tambahkan kode berikut di dalam kelas `HomeState`.

```
List<Music> musics = [];  
List<Video> videos = [];
```

8. Agar kedua state tersebut dapat menyimpan data musik dan video sesaat ketika halaman Home siap ditampilkan, anda perlu menambahkan fungsi `getData()` yang dipanggil pada fungsi `initState()`.

```
@override  
void initState() {  
  super.initState();  
  getData();  
}  
  
Future<void> getData() async {  
  await Repository.getMusics();  
  await Repository.getVideos();  
  setState(() {  
    musics = Repository.musics;  
    videos = Repository.videos;  
  });  
}
```

Fungsi `getData()` akan mengambil data musik dan video dari repository, kemudian menyimpan data yang didapat ke dalam state `musics` dan `videos`.

9. Kembali pada kode widget `ListView.separated [I]`, tambahkan properti **itemCount** dengan nilai `videos.length`.
10. Pada bagian **itemBuilder**, isi dengan sebuah fungsi yang mengembalikan widget untuk menampilkan komponen `CoverVideoCard`. Di bawah ini adalah nilai yang perlu anda masukkan pada properti **itemBuilder**.

```
(_, i) => Padding(  
  padding: EdgeInsets.only(bottom: i == 4 ? 16 : 0),  
  child: CoverVideoCard(  
    video: videos[i],  
  ),  
)
```

11. Berikut kode akhir dari widget ListView.separated [I].

```

ListView.separated(
  key: const Key('video_list_view'),
  physics: const NeverScrollableScrollPhysics(),
  shrinkWrap: true,
  itemCount: videos.length,
  itemBuilder: (_, i) => Padding(
    padding: EdgeInsets.only(bottom: i == 4 ? 16 : 0),
    child: CoverVideoCard(
      video: videos[i],
    ), // CoverVideoCard
  ), // Padding
  separatorBuilder: (context, index) => const SizedBox(
    height: 12,
  ), // SizedBox
), // ListView.separated

```

12. ListView yang anda modifikasi sebelumnya adalah kode untuk menampilkan list video saja. Untuk menampilkan list musik, anda perlu membuat ListView.separated baru sebagai child dari widget SizedBox [F]. Sesuaikan juga setiap propertinya berdasarkan spesifikasi di bawah ini.

Widget	Properti	Nilai
ListView.separated	key	const Key('music list view')
	scrollDirection	Axis.horizontal
	itemCount	musics.length
	itemBuilder	(_, i) { final item = musics[i]; return Padding( padding: EdgeInsets.only( left: i == 0 ? 15 : 0, right: i == musics.length - 1 ? 15 : 0, ), child: CoverMusicCard(music: item), ); }
separatorBuilder	(_, i) => const SizedBox( width: 20, )	

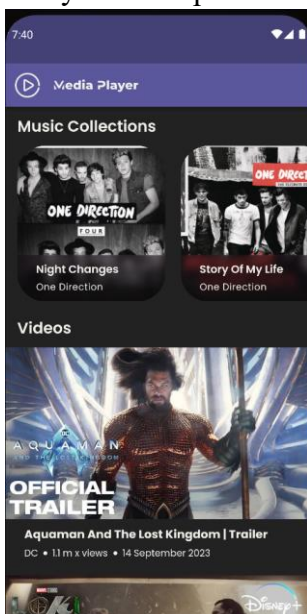
13. Berikut adalah kode akhir dari widget `SizeBox` [F]

```
), // ListTile
)-SizeBox(
  height: 200,
  child: ListView.separated(
    key: const Key('music_list_view'),
    scrollDirection: Axis.horizontal,
    itemBuilder: (_, i) {
      final item = musics[i];
      return Padding(
        padding: EdgeInsets.only(
          left: i == 0 ? 15 : 0,
          right: i == musics.length - 1 ? 15 : 0,
        ), // EdgeInsets.only
        child: CoverMusicCard(music: item),
      ); // Padding
    },
    separatorBuilder: (_, i) => const SizeBox(
      width: 20,
    ), // SizeBox
    itemCount: musics.length,
  ), // ListView.separated
), // SizeBox
```

14. Sebelum menjalankannya pada emulator, anda harus mengubah kode pada file `main_pages.dart`. Untuk membuat tampilan Home benar-benar ditampilkan pada emulator sesaat setelah `SplashScreen` muncul dan animasinya selesai, ubah widget kembalian dari widget function `MainRoute.home` dari yang sebelumnya `Placeholder()` menjadi `Home()`.

```
Map<String, Widget Function(BuildContext)> mainPages = {
  MainRoute.splash: (context) => const SplashScreen(),
  MainRoute.home: (context) => const Home(),
  MainRoute.musicPlayer: (context) => const Placeholder(),
```

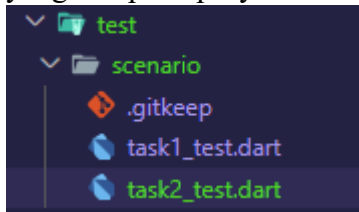
15. Coba jalankan proyek anda pada emulator dengan menekan tombol run yang ada pada fungsi main di `main.dart`. Berikut adalah tampilan emulator jika anda telah berhasil menyusun tampilan Home.



Anda juga dapat melihat tampilan keseluruhan aplikasinya [di sini](#).

## Langkah Verifikasi Kode

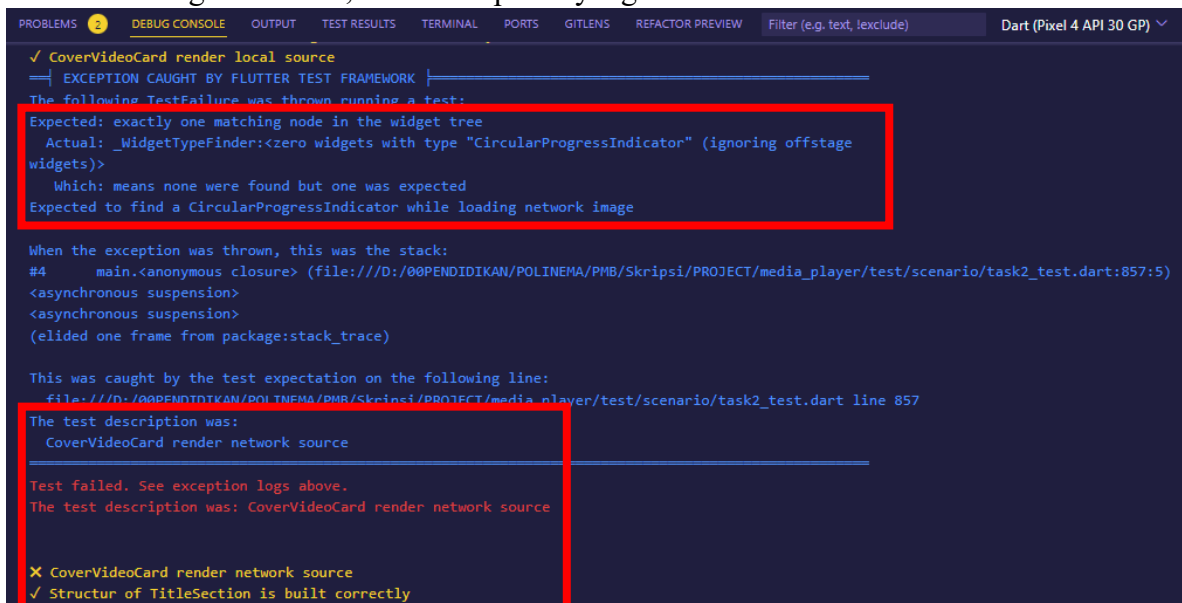
1. Unduh test file pada tautan [Test File Tugas 2](#). Letakkan file ke dalam folder test>scenario yang ada pada proyek.



2. Buka file **task2\_test.dart**.

```
test > scenario > task2_test.dart > main
1 > import 'dart:ui'; ...
19
Run | Debug
20 void main() {
Run | Debug
21 > testWidgets('Structur of CustomAppBar widget is built correctly', ...
133
Run | Debug
2134 > testWidgets('CustomAppBar widget display provided leading and title wid
228
Run | Debug
229 > testWidgets('Structur of CoverMusicCard widget is built correctly', ...
433
Run | Debug
434 > testWidgets('CoverMusicCard render local source', ...
493
Run | Debug
494 > testWidgets('CoverMusicCard render network source', ...
560
```

3. Tekan tombol Run yang ada pada fungsi main dan tunggu proses hingga selesai.
4. Jika anda mengalami error, coba lihat pesan yang diberikan.



5. “Expected to find a CircularProgressIndicator while loading network image”. Maksud dari pesan error tersebut adalah komponen CoverVideoCard seharusnya menampilkan

CircularProgressIndicator saat proses memuat (loading) gambar dari internet. Sehingga saat anda mengalami error yang serupa, tambahkan properti `progressIndicatorBuilder` pada widget `CachedNetworkImage` sesuai dengan spesifikasi berikut.

Widget	Properti	Nilai
<code>CachedNetworkImage</code>	<code>progressIndicatorBuilder</code>	<pre>(context, url, progress) =&gt;   const Center(     child:       CircularProgressIndicator(         color:           MainColor.purple5A579C,       ),   )</pre>

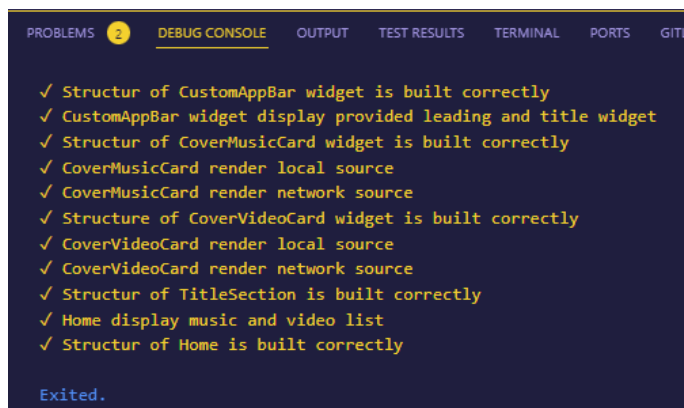
```

EdgeInsets(
  width: double.infinity,
  height: MediaQuery.sizeOf(context).width * 9 / 16,
  child: video.sourceType == "local"
    ? Image.asset(
        video.coverPath!,
        fit: BoxFit.cover,
      ) // Image.asset
    : ClipRRect(
        child: CachedNetworkImage(
          imageUrl: video.coverPath!,
          progressIndicatorBuilder: (context, url, progress) =>
            const Center(
              child: CircularProgressIndicator(
                color: MainColor.purple5A579C,
              ), // CircularProgressIndicator
            ), // Center
          fit: BoxFit.cover,
        ), // CachedNetworkImage
      ), // ClipRRect
    ), // EdgeInsets
  ), // SizedBox

```

\*Properti `progressIndicatorBuilder` pada `CachedNetworkImage` berfungsi untuk mengatur widget yang akan ditampilkan saat target [imageUrl] sedang dimuat.

- Jalankan kembali test file dengan menekan tombol Run yang ada pada fungsi main dan tunggu proses hingga selesai.
- Jika testing berhasil, maka akan muncul hasil berikut pada tab Test Result Visual Studio Code anda.



```

PROBLEMS 2  DEBUG CONSOLE  OUTPUT  TEST RESULTS  TERMINAL  PORTS  GITL
✓ Structur of CustomAppBar widget is built correctly
✓ CustomAppBar widget display provided leading and title widget
✓ Structur of CoverMusicCard widget is built correctly
✓ CoverMusicCard render local source
✓ CoverMusicCard render network source
✓ Structure of CoverVideoCard widget is built correctly
✓ CoverVideoCard render local source
✓ CoverVideoCard render network source
✓ Structur of TitleSection is built correctly
✓ Home display music and video list
✓ Structur of Home is built correctly

Exited.

```