

Modul Pratikum Pertemuan 5: Material Design & Animasi di Jetpack Compose

1. Material Design di Jetpack Compose

a. Apa itu Material Design?

Material Design adalah bahasa desain dari Google yang menyediakan panduan tampilan agar aplikasi konsisten, modern, dan mudah digunakan.

Jetpack Compose sudah memiliki implementasi **Material Design Components** bawaan sehingga developer tidak perlu membangun komponen UI dari nol.

b. MaterialTheme

```
MaterialTheme(  
    colorScheme = /* ...  
    typography = /* ...  
    shapes = /* ...  
) {  
    // M3 app content  
}
```

Di dalamnya terdapat tiga aspek utama:

1. **Color Scheme** → menentukan warna utama, sekunder, background, dll. kumpulan warna utama. Setiap warna ini terkait dengan palet tonal 13 tone, yang digunakan oleh komponen Material 3. Misalnya, berikut adalah skema warna untuk tema terang

Membuat skema warna:

Color.kt berisi warna tema Anda dengan semua peran yang ditentukan untuk warna tema terang dan gelap.

```
val md_theme_light_primary = Color(0xFF476810)  
val md_theme_light_onPrimary = Color(0xFFFFFFFF)
```

```

val md_theme_light_primaryContainer = Color(0xFFC7F089)
// ..
// ..

val md_theme_dark_primary = Color(0xFFACD370)
val md_theme_dark_onPrimary = Color(0xFF213600)
val md_theme_dark_primaryContainer = Color(0xFF324F00)
// ..

```

Theme.kt berisi penyiapan skema warna terang dan gelap serta tema aplikasi.

```

private val LightColorScheme = lightColorScheme(
    primary = md_theme_light_primary,
    onPrimary = md_theme_light_onPrimary,
    primaryContainer = md_theme_light_primaryContainer,
    // ..
)

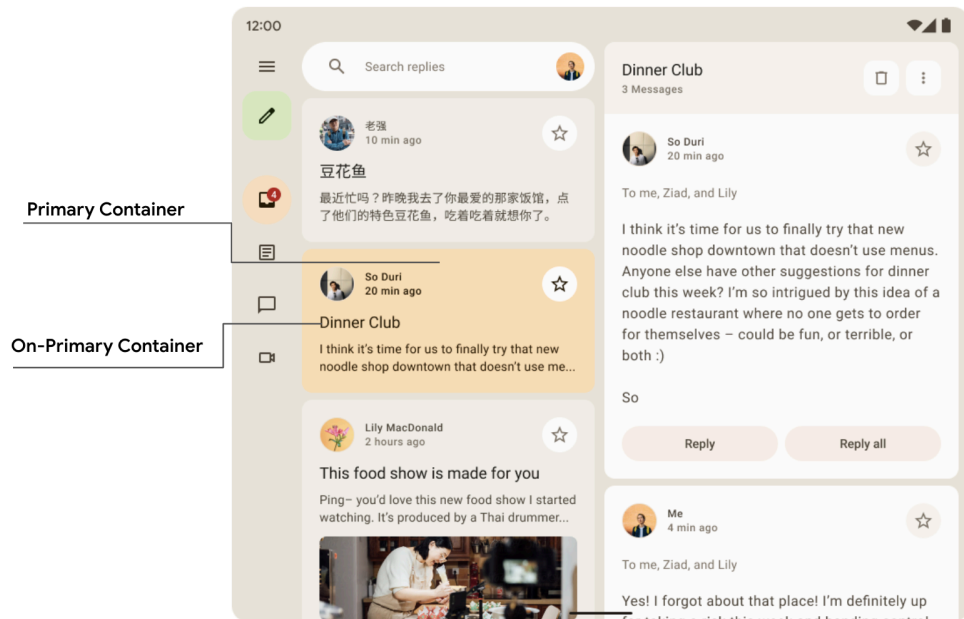
private val DarkColorScheme = darkColorScheme(
    primary = md_theme_dark_primary,
    onPrimary = md_theme_dark_onPrimary,
    primaryContainer = md_theme_dark_primaryContainer,
    // ..
)

@Composable
fun ReplyTheme(
    darkTheme: Boolean = isSystemInDarkTheme(),
    content: @Composable () -> Unit
) {
    val colorScheme =
        if (!darkTheme) {
            LightColorScheme
        } else {
            DarkColorScheme
        }
    MaterialTheme(
        colorScheme = colorScheme,
        content = content
    )
}

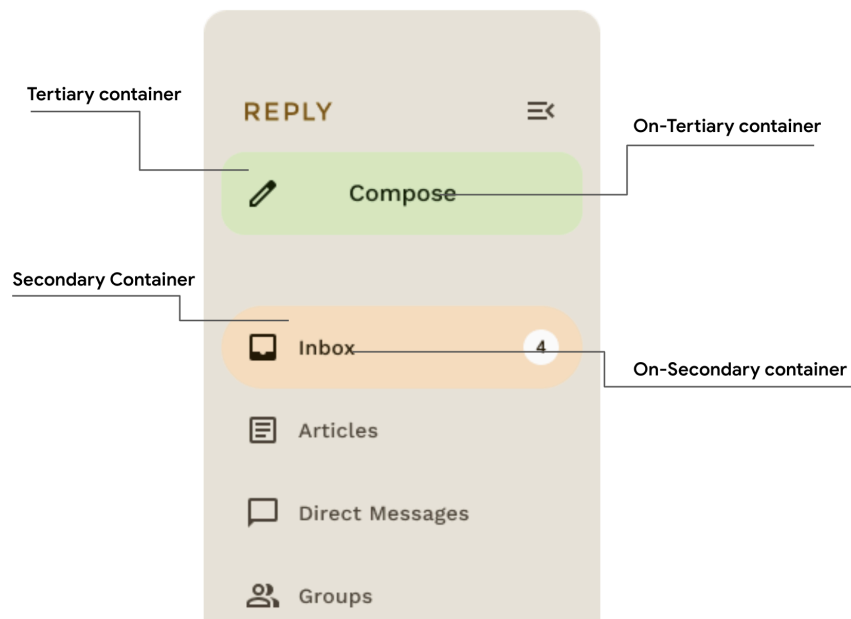
```

Penggunaan warna:

```
Text(  
    text = "Hello theming",  
    color = MaterialTheme.colorScheme.primary  
)
```



```
Card(  
    colors = CardDefaults.cardColors(  
        containerColor =  
            if (isSelected)  
                MaterialTheme.colorScheme.primaryContainer  
            else  
                MaterialTheme.colorScheme.surfaceVariant  
    )  
) {  
    Text(  
        text = "Dinner club",  
        style = MaterialTheme.typography.bodyLarge,  
        color =  
            if (isSelected)  
                MaterialTheme.colorScheme.onPrimaryContainer  
            else MaterialTheme.colorScheme.onSurface,  
    )  
}
```



2. **Typography** → gaya teks (ukuran, ketebalan, jenis font).

Skala tipografi default untuk Desain Material 3

Display Large	Roboto 57/64 . 0
Display Medium	Roboto 45/52 . 0
Display Small	Roboto 36/44 . 0
Headline Large	Roboto 32/40 . 0
Headline Medium	Roboto 28/36 . 0
Headline Small	Roboto 24/32 . 0
Title Large	NEW - Roboto Medium 22/28 . 0
Title Medium	Roboto Medium 16/24 . +0.15
Title Small	Roboto Medium 14/20 . +0.1
Body Large	Roboto 16/24 . +0.5
Body Medium	Roboto 14/20 . +0.25
Body Small	Roboto 12/16 . +0.4
Label Large	Roboto Medium 14/20 . +0.1
Label Medium	Roboto Medium 12/16 . +0.5
Label Small	NEW - Roboto Medium 11/16 . +0.5

Headline 1	H1/Roboto/Light
Headline 2	H2/Roboto/Light/60px
Headline 3	H3/Roboto/Regular/48px
Headline 4	H4/Roboto/Regular/34px
Headline 5	H5/Roboto/Regular/24px
Headline 6	H6/Roboto/Medium/20px
Subtitle 1	Subtitle 1/Roboto/Regular/16px
Subtitle 2	Subtitle 2/Roboto/Medium/14px
Body 1	Body 1/Roboto/Regular/16px
Body 2	Body 2/Roboto/Regular/14px
Caption	Caption/Roboto/Regular/12px
Button	BUTTON/ROBOTO/MEDIUM/14PX
Overline	OVERLINE/ROBOTO/REGULAR/10PX

Definisikan tipografi:

```
val replyTypography = Typography(
    titleLarge = TextStyle(
        fontWeight = FontWeight.SemiBold,
        fontSize = 22.sp,
        lineHeight = 28.sp,
        letterSpacing = 0.sp
    ),
    titleMedium = TextStyle(
        fontWeight = FontWeight.SemiBold,
        fontSize = 16.sp,
        lineHeight = 24.sp,
        letterSpacing = 0.15.sp
    ),
    // ..
)
```

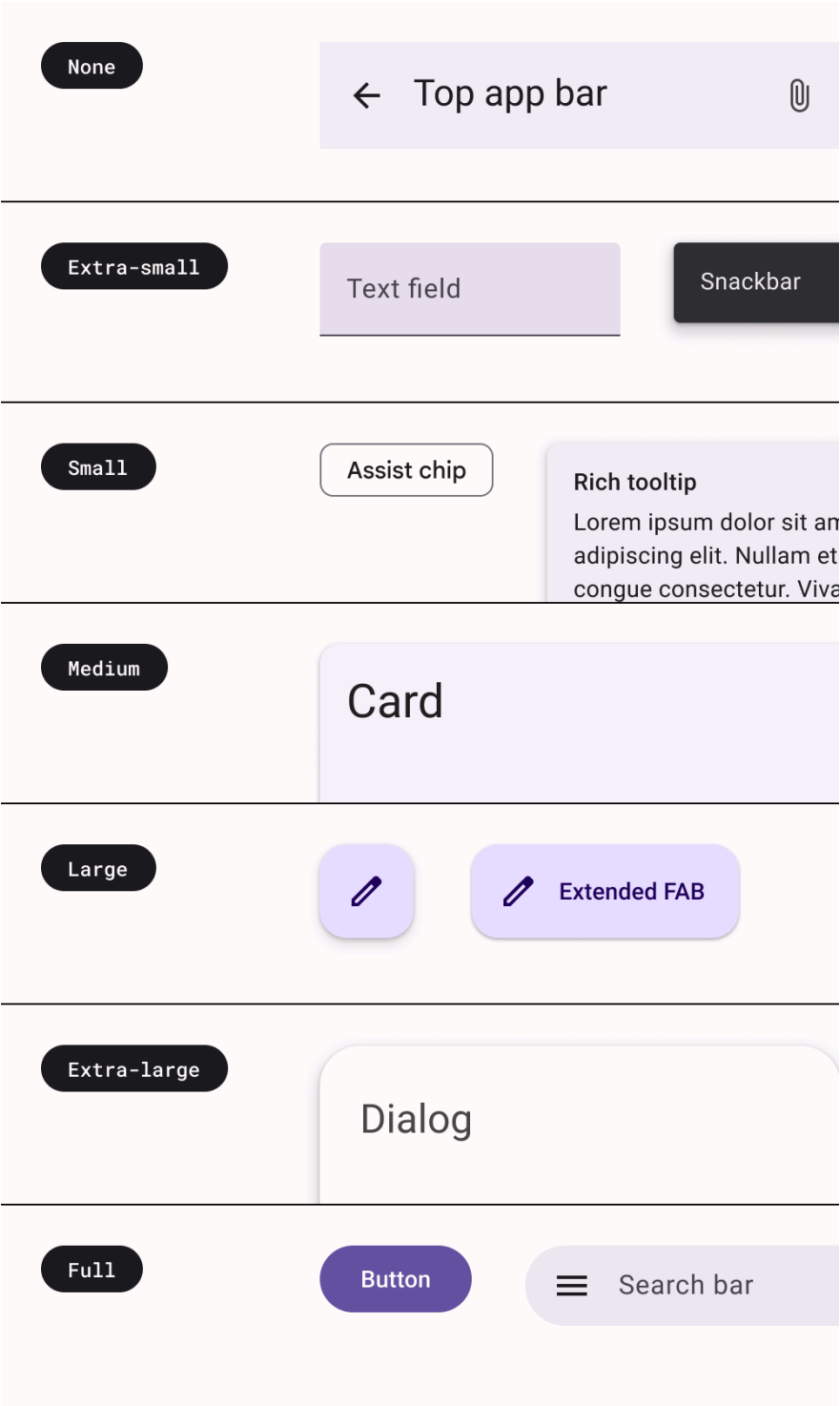
Cara gunakan text styles:

```
Text(
    text = "Hello M3 theming",
```

```
        style = MaterialTheme.typography.titleLarge
    )
    Text(
        text = "you are learning typography",
        style = MaterialTheme.typography.bodyMedium
    )
}
```

3. **Shapes** → bentuk elemen UI (rounded corner, cut corner, dsb).

```
val replyShapes = Shapes(
    extraSmall = RoundedCornerShape(4.dp),
    small = RoundedCornerShape(8.dp),
    medium = RoundedCornerShape(12.dp),
    large = RoundedCornerShape(16.dp),
    extraLarge = RoundedCornerShape(24.dp)
)
```



Cara pakai shape:

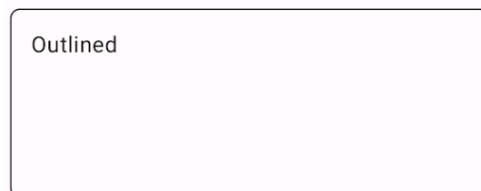
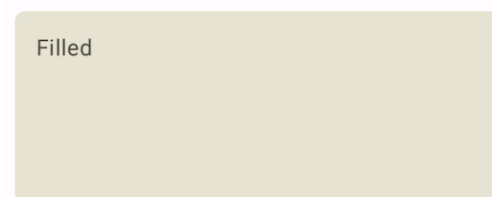
```
Card(shape = MaterialTheme.shapes.medium) { /* card
content */ }
FloatingActionButton(
    shape = MaterialTheme.shapes.large,
    onClick = {
    }
) {
    /* fab content */
}
```

c. Komponen Material yang Umum Digunakan

1. Card

Wadah dengan bayangan (elevation) untuk menampilkan konten terpisah.

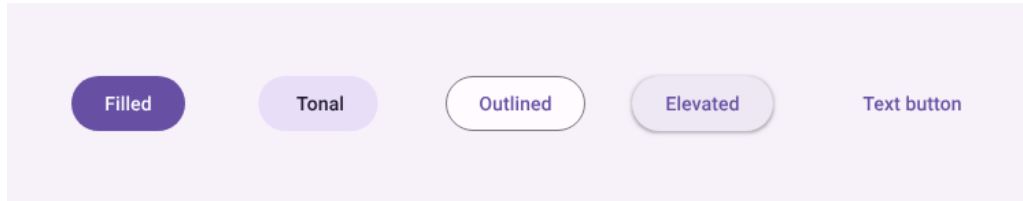
Sering dipakai di daftar, tampilan produk, atau konten berita.



2. Button

Elemen interaktif untuk aksi utama pengguna.

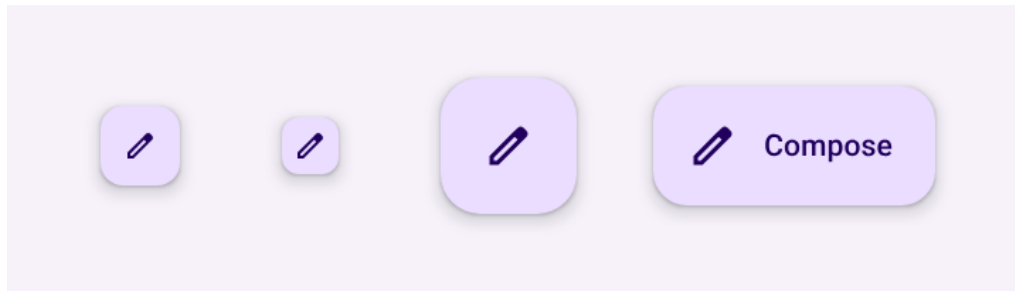
Bisa berupa Button Filled, Tonal, OutlinedButton, Elevated atau TextButton.



3. Floating Action Button (FAB)

Tombol melayang yang menonjol untuk aksi utama di layar (misalnya “Tambah”).

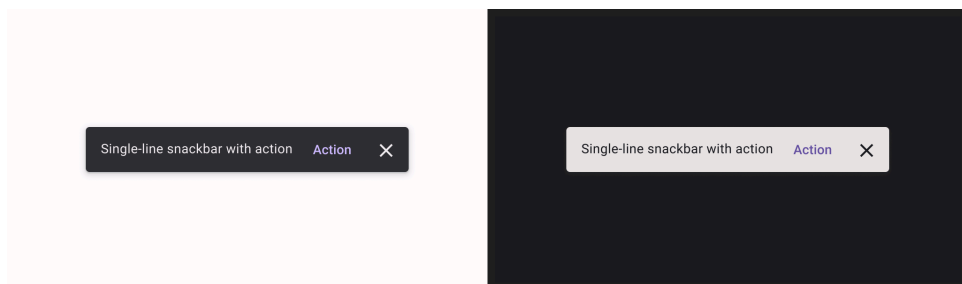
Biasanya ditempatkan di kanan bawah layar.



4. Snackbar

Komponen notifikasi singkat di bagian bawah layar.

Cocok untuk memberi feedback setelah aksi tertentu (misalnya “Data berhasil disimpan”).



Source Code Material Theme:

<https://github.com/Nopallse/material-theme.git>

2. Animasi di Jetpack Compose

a. Konsep Animasi di Compose

UI di Compose adalah fungsi dari **state**.

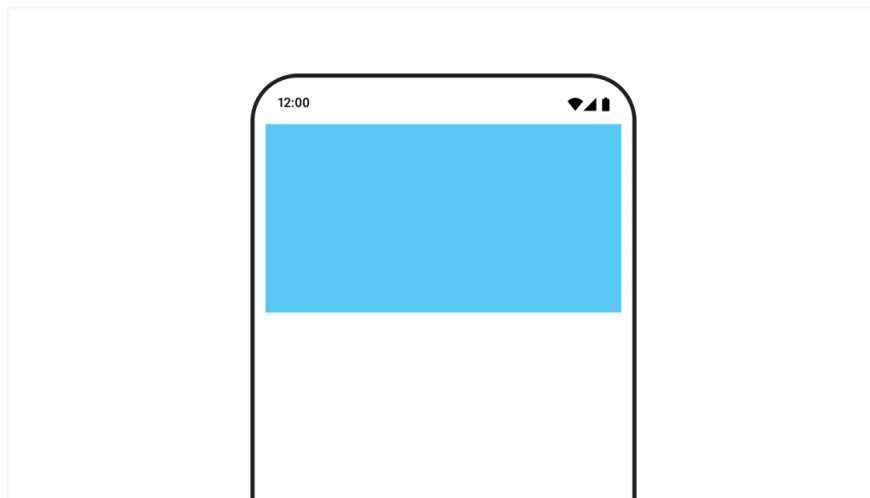
Jika state berubah → UI akan otomatis *recompose*.

Animasi memungkinkan transisi perubahan state menjadi lebih halus dan interaktif.

b. Jenis Animasi Dasar

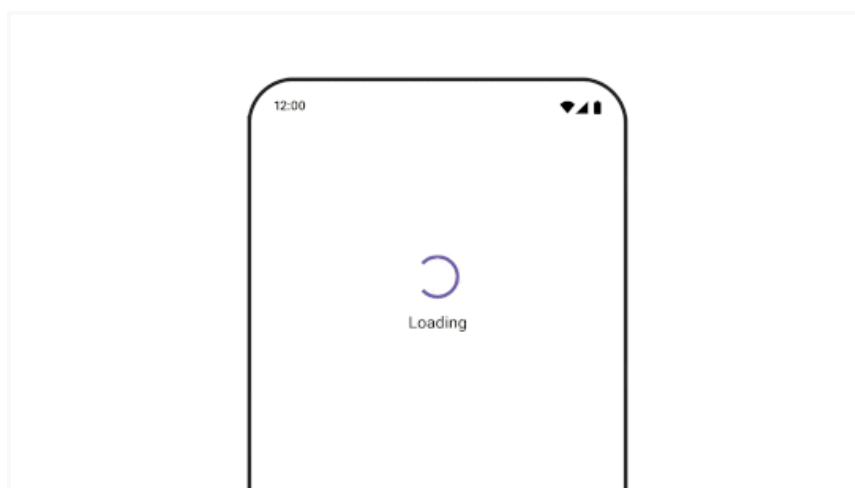
1. `animateContentSize` (Menganimasikan perubahan ukuran konten)

untuk menjalankan animasi perubahan ukuran secara otomatis



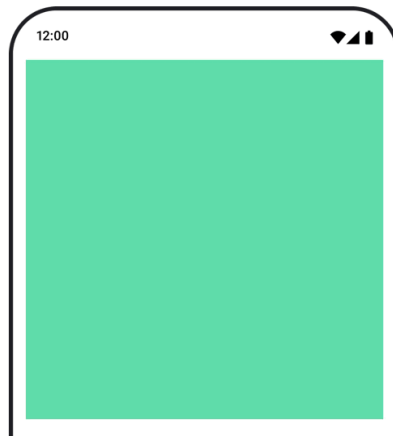
2. `AnimatedContent` (Menganimasikan antar-composable yang berbeda)

Gunakan `AnimatedContent` untuk menganimasikan antar-composable yang memiliki konten berbeda.



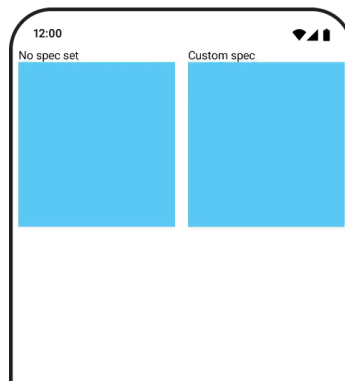
3. `animate*AsState` (**Menganimasikan satu nilai**)

Gunakan fungsi `animate*AsState` untuk menganimasikan properti individual



4. `Transition` (**Menganimasikan beberapa nilai bersama-sama**)

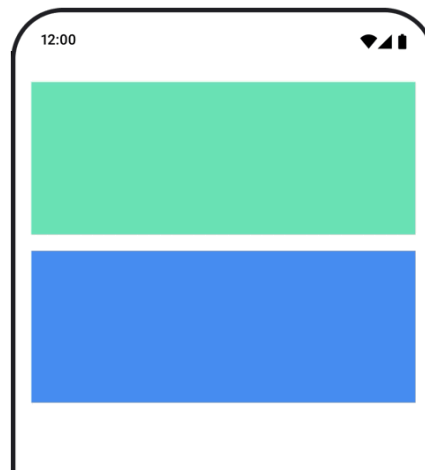
menganimasikan beberapa nilai sekaligus.



5. **AnimatedVisibility** (Menganimasikan muncul dan menghilang)

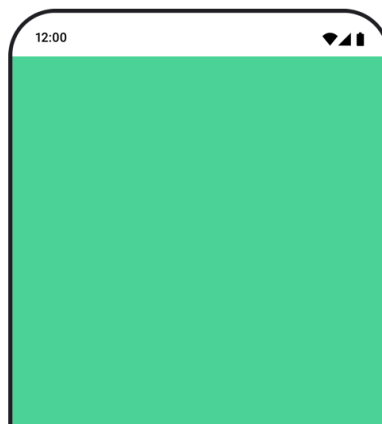
Digunakan untuk mengatur muncul atau hilangnya sebuah elemen UI dengan animasi.

Cocok untuk efek show/hide teks, gambar, atau komponen lainnya.



6. **InfiniteTransition** (Menganimasikan properti tanpa batas)

untuk menganimasikan properti secara terus-menerus.



c. Manfaat Animasi

Membuat aplikasi terasa lebih **hidup** dan **responsif**.

Memberi **feedback visual** ke pengguna ketika terjadi perubahan.

Menarik perhatian pengguna ke elemen penting.

Link repo:

<https://github.com/Nopallse/animation.git>

Tugas Praktikum Pertemuan 5

Deskripsi

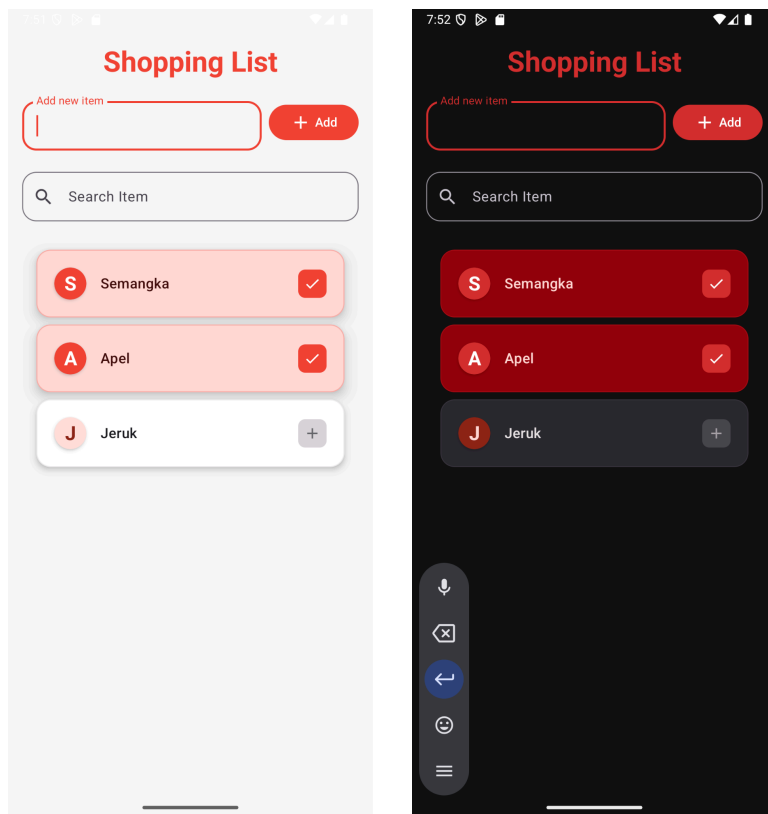
Lanjutkan aplikasi **Shopping List** yang sudah dibuat pada pertemuan sebelumnya dengan menambahkan **Material Design** dan **Animasi Sederhana** agar daftar belanja terlihat lebih menarik dan interaktif.

Gunakan project **Shopping List** minggu sebelumnya sebagai starter project.

Instruksi

1. **Sesuaikan warna aplikasi** dengan tema masing-masing:
 - Gunakan **MaterialTheme** atau custom color.
 - Atur warna untuk **tema terang (light theme)** dan juga **tema gelap (dark theme)** agar aplikasi tetap nyaman digunakan di kedua mode.
2. **Tampilkan setiap item dalam Card:**
 - Gunakan komponen **Card** dari Material Design.
 - Tambahkan **ikon** di dalam Card (misalnya ikon belanja atau tanda centang).
3. **Tambahkan Animasi pada aplikasi:**
 - Gunakan **animasi warna atau elevasi** ketika item di-*klik*.
 - Gunakan **AnimatedVisibility** (atau animasi lain) ketika item baru masuk ke dalam list.
4. **Pastikan UI tetap responsif** dan mengikuti gaya dari **MaterialTheme**.

Hasil



Source Code

1. Shape.kt

```
package com.example.shoppinglist.ui.theme

import androidx.compose.material3.Shapes
import androidx.compose.ui.unit.dp

val Shapes = Shapes(
    small = androidx.compose.foundation.shape.RoundedCornerShape(12.dp),
    medium = androidx.compose.foundation.shape.RoundedCornerShape(12.dp),
    large = androidx.compose.foundation.shape.RoundedCornerShape(12.dp)
)
```

2. [Color.kt](#)
3. [Theme.kt](#)
4. [Type.kt](#)
5. [ShoppingList.kt](#)