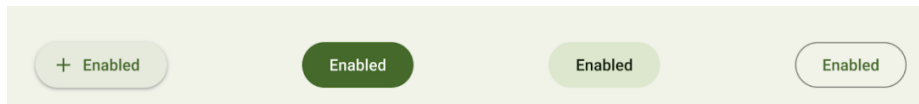


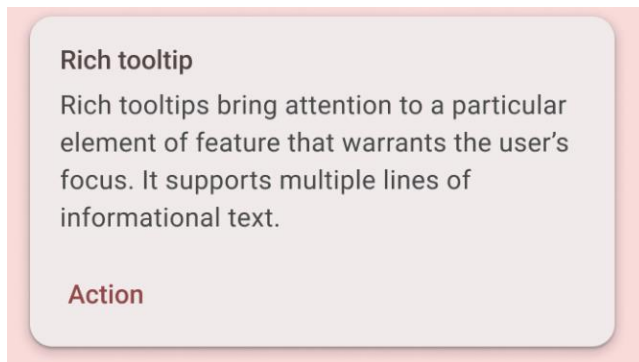
Tata Letak Dasar

Antarmuka Pengguna (UI)

Antarmuka pengguna (UI) aplikasi adalah tampilan visual yang Anda lihat di layar: teks, gambar, tombol, dan berbagai jenis elemen lainnya, serta cara elemen tersebut diletakkan di layar.



Tombol yang dapat diklik



Pesan teks di dalam Kartu



Kolom input teks

Jetpack Compose

Jetpack Compose adalah toolkit modern untuk membangun UI Android. Compose menyederhanakan dan mempercepat pengembangan UI di Android dengan lebih sedikit kode, alat yang canggih, dan kemampuan Kotlin yang intuitif. Compose dapat membantu Anda membangun UI dengan menentukan sekumpulan fungsi, yang disebut fungsi composable, yang mengambil data dan menjelaskan elemen UI.

Fungsi composable

Fungsi composable adalah elemen dasar penyusun UI di Compose. Fungsi composable:

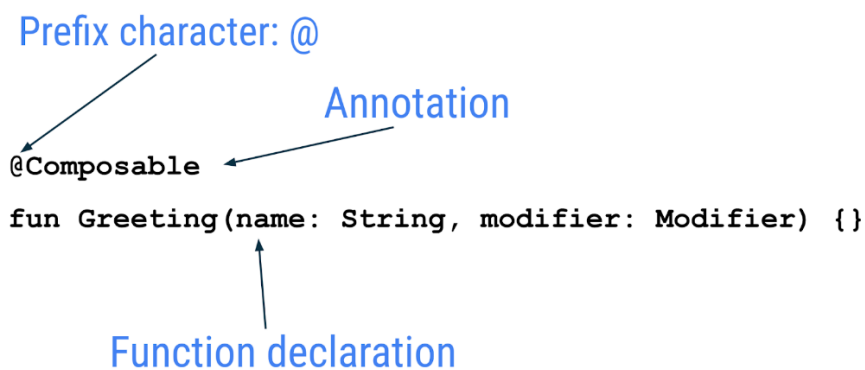
- Menggambarkan beberapa bagian UI Anda.
- Tidak menghasilkan apa pun.
- Mengambil beberapa input dan menghasilkan elemen yang ditampilkan di layar.

Anotasi

Anotasi adalah cara untuk melampirkan informasi tambahan ke kode. Informasi ini membantu alat seperti compiler Jetpack Compose, dan developer lain memahami kode aplikasi.

Anotasi diterapkan dengan menambahkan awalan nama (anotasi) dengan karakter @ di awal deklarasi yang Anda anotasi. Berbagai elemen kode, seperti properti, fungsi, dan class, dapat dianotasi.

Diagram berikut adalah contoh fungsi yang dianotasi:



Contoh fungsi composable

Fungsi composable ini dianotasi dengan anotasi `@Composable`. Semua fungsi composable harus memiliki anotasi ini. Anotasi ini memberi tahu compiler Compose bahwa fungsi ini dimaksudkan untuk mengonversi data menjadi UI.

Cuplikan kode ini adalah contoh dari fungsi composable sederhana yang meneruskan data (parameter fungsi `name`) dan menggunakannya untuk merender elemen teks di layar.

```
@Composable
fun Greeting(name: String) {
    Text(text = "Hello $name!")
}
```

Nama fungsi composable

Fungsi compose yang tidak menghasilkan apa pun dan menyertakan anotasi `@Composable` HARUS diberi nama menggunakan Pascal case. Pascal case mengacu pada konvensi penamaan yang menggunakan huruf kapital untuk huruf pertama setiap kata dalam kata majemuk.

Fungsi Compose:

- HARUS kata benda: `DoneButton()`
- BUKAN kata kerja atau frasa kata kerja: `DrawTextField()`
- BUKAN preposisi yang dibuat menjadi kata benda: `TextFieldWithLink()`
- BUKAN kata sifat: `Bright()`
- BUKAN kata keterangan: `Outside()`
- Kata benda DAPAT diawali dengan kata sifat deskriptif: `RoundIcon()`

Pratinjau

Pratinjau (preview) digunakan untuk menampilkan hasil tampilan dengan tidak menjalankan di perangkat. Pratinjau menggunakan anotasi `@Preview` dan `@Composable`.

```
@Composable
fun GreetingText(message: String, modifier: Modifier = Modifier) {
    Text(
        text = message
    )
}

@Preview(showBackground = true)
@Composable
fun BirthdayCardPreview() {
    HappyBirthdayTheme {
        GreetingText(message = "Happy Birthday Sam!")
    }
}
```

Hasil preview

BirthdayCardPreview

Happy Birthday Sam!

Mengubah ukuran font

[Piksel skalabel \(SP\)](#) adalah satuan ukuran untuk ukuran font. Elemen UI di aplikasi Android menggunakan dua unit pengukuran yang berbeda: [piksel kepadatan mandiri \(DP\)](#) yang Anda gunakan nanti untuk tata letak, dan piksel skalabel (SP).

```
import androidx.compose.ui.unit.sp
...

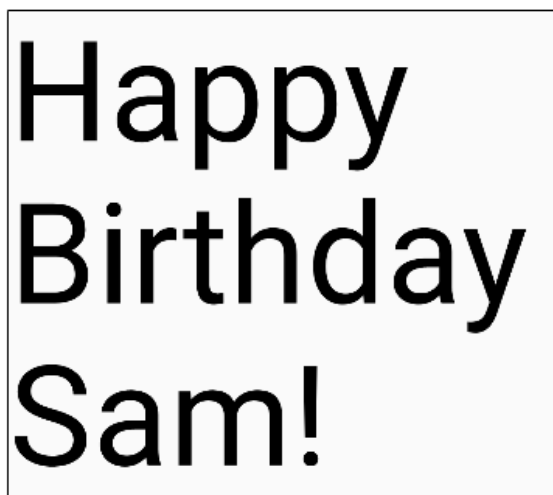
Text(
    text = message,
    fontSize = 100.sp
)
```

menyertakan tinggi baris.

```
Text(
    text = message,
    fontSize = 100.sp,
    lineHeight = 116.sp,
)
```

Hasil

BirthdayCardPreview



Menambahkan elemen teks lainnya

```
@Composable
fun GreetingText(message: String, from: String, modifier: Modifier =
Modifier) {
    Text(
        text = message,
        fontSize = 100.sp,
        lineHeight = 116.sp,
    )
    Text(
        text = from,
```

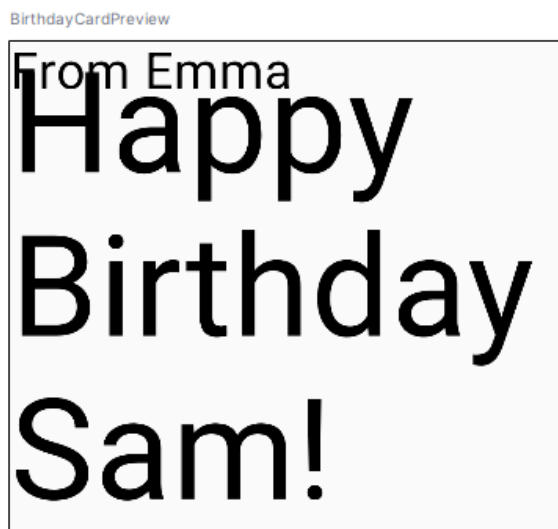
```

        fontSize = 36.sp
    )
}

@Preview(showBackground = true)
@Composable
fun BirthdayCardPreview() {
    HappyBirthdayTheme {
        GreetingText(message = "Happy Birthday Sam!", from = "From Emma")
    }
}

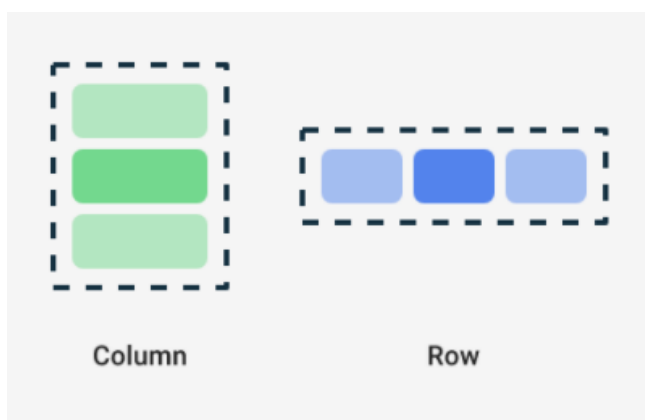
```

Hasil



Mengatur elemen teks dalam baris dan kolom

Tiga elemen tata letak standar dan dasar di Compose adalah composable **Column**, **Row**, dan **Box**. Anda dapat mempelajari composable **Box** lebih lanjut di codelab berikutnya.

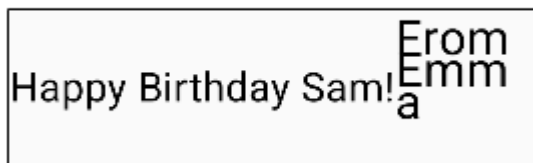


Mengatur elemen teks dalam baris

```
import androidx.compose.foundation.layout.Row
...
@Composable
fun GreetingText(message: String, from: String, modifier: Modifier =
Modifier) {
    Row {
        Text(
            text = message,
            fontSize = 30.sp,
            lineHeight = 116.sp,
        )
        Text(
            text = from,
            fontSize = 36.sp
        )
    }
}
```

Hasil

BirthdayCardPreview



Mengatur elemen teks dalam kolom

```
import androidx.compose.foundation.layout.Column

@Composable
fun GreetingText(message: String, from: String, modifier: Modifier =
Modifier) {
    Column {
        Text(
            text = message,
            fontSize = 100.sp,
            lineHeight = 116.sp,
        )
        Text(
            text = from,
            fontSize = 36.sp
        )
    }
}
```

Happy
Birthday
Sam!
From Emma

Menambahkan ucapan ke aplikasi

Fungsi `setContent()` dalam fungsi `onCreate()` diubah menjadi seperti berikut.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContent {
        HappyBirthdayTheme {
            Surface(
                modifier = Modifier.fillMaxSize(),
                color = MaterialTheme.colorScheme.background
            ) {
                GreetingText(
                    message = "Happy Birthday Sam!",
                    from = "From Emma",
                    modifier = Modifier.padding(8.dp)
                )
            }
        }
    }
}
```



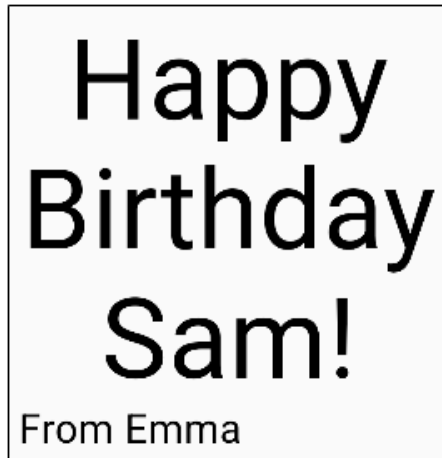
Menyejajarkan ucapan ke tengah

```
Column(  
    verticalArrangement = Arrangement.Center,  
    modifier = modifier.padding(8.dp)  
) {  
    // ...  
}
```

sejajarkan teks ucapan ke tengah menggunakan `textAlign`.

```
Text(  
    text = message,  
    fontSize = 100.sp,  
    lineHeight = 116.sp,  
    textAlign = TextAlign.Center  
)
```


BirthdayCardPreview



Tambahkan padding dan sejajarkan di sebelah kanan.

```
Text(  
  text = from,  
  fontSize = 36.sp,  
  modifier = Modifier  
    .padding(16.dp)  
    .align(alignment = Alignment.End)  
)
```

BirthdayCardPreview

