

MODUL PRAKTIKUM PEMROGRAMAN UNTUK PERANGKAT BERGERAK 1

Indra Azimi, S.T., M.T.

Reza Budiawan, S.T., M.T., OCA

Dwiko Indrawansyah, A.Md.Kom.



D3 Rekayasa Perangkat Lunak Aplikasi
Telkom University

Daftar Isi

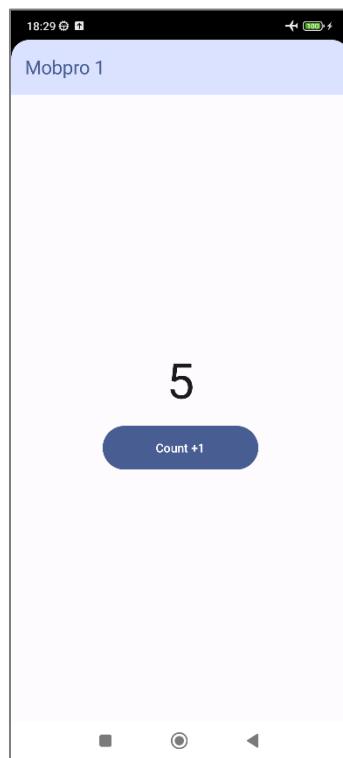
Modul 02: Build Your First App	1
1. Overview	1
2. Getting Started.....	1
3. Task	7
3.1. Menambahkan Top App Bar	7
3.2. Membuat Tampilan Counter	11
3.3. Memberi Aksi Pada Tombol Saat Diklik	14
3.4. Mendukung Bahasa Lokal (Indonesia).....	14
4. Summary	16
5. Self-Reflection.....	16
6. Challenge.....	17

Modul 02: Build Your First App

1. Overview

Pada modul kali ini, kita akan membuat sebuah aplikasi sederhana berupa counter app. Aplikasi ini menampilkan sebuah angka dan sebuah tombol di tengah layar. Angka yang muncul akan bertambah satu setiap kali tombol ditekan. Aplikasi yang akan kita buat dapat dilihat kode lengkapnya di <https://github.com/indraazimi/mobpro1-compose> branch **02-first-app**.

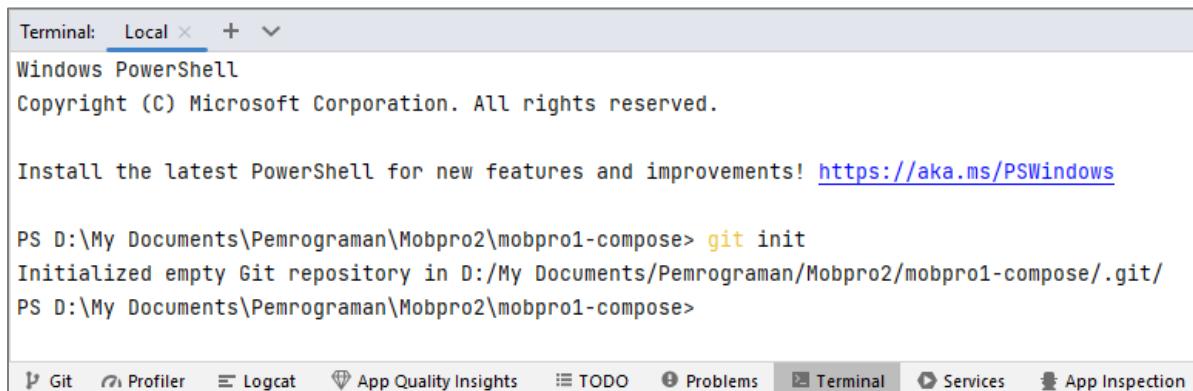
Tampilan aplikasi yang akan dibuat saat ini adalah sebagai berikut:



2. Getting Started

Kali ini kita akan menggunakan version control Git. Jadi jika belum punya Git, unduh terlebih dahulu Git dari [sini](#), lalu install di komputer/laptop masing-masing. Kemudian jalankan Android Studio, lalu buka project pertama dari **modul minggu lalu** yang bernama Hello Android! Pastikan project tersebut masih dapat dijalankan dengan baik di smartphone/emulator sebelum lanjut mengerjakan modul ini.

Berikutnya, jadikan project Android Studio yang sudah dibuat sebelumnya agar menjadi project yang menggunakan versioning Git. Caranya, buka Terminal yang terletak di bagian bawah Android Studio. Di terminal tersebut, ketikkan perintah “git init” lalu tekan Enter.



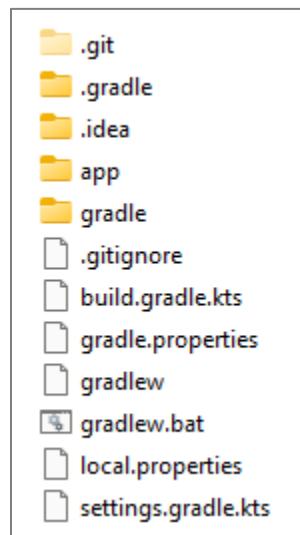
```
Terminal: Local × + ▾
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose> git init
Initialized empty Git repository in D:/My Documents/Pemrograman/Mobpro2/mobpro1-compose/.git/
PS D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose>
```

The screenshot shows a Windows PowerShell terminal window within the Android Studio interface. The terminal title is "Local". The output shows the command "git init" being run in the directory "D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose". The response from the command is "Initialized empty Git repository in D:/My Documents/Pemrograman/Mobpro2/mobpro1-compose/.git/". Below the terminal, the Android Studio navigation bar is visible, featuring icons for Git, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, and App Inspection.

Jika sudah terdapat keterangan “Initialized empty Git repository in ...” artinya proses inisialisasi Git telah berhasil. Untuk memastikannya, buka folder project menggunakan File Explorer. Di sana akan terlihat ada hidden folder bernama “.git” seperti pada gambar. Jika folder tersebut tidak ada, berarti kamu belum mengatur File Explorer agar menampilkan hidden folder.



Sampai di sini, project Android Studio yang kita buat telah dimungkinkan untuk dilakukan operasi berupa commit, push, dan lainnya. Operasi commit memiliki arti yang mirip dengan checkpoint pada sebuah game. Kita dapat kembali ke titik commit tersebut jika ada kode yang tidak sesuai harapan saat proses membangun aplikasi. Sedangkan operasi push merupakan operasi untuk mengirimkan kode kita agar tersimpan di repository online seperti Github.

Langkah selanjutnya adalah menyiapkan repository di Github untuk menyimpan project kita ini. Buat akun di [Github](#) terlebih dahulu jika masih belum punya. Lalu buka link Github Assignment yang disediakan oleh dosen untuk membuat repository praktikum. Setelah repository jadi, carilah perintah yang diberikan tanda merah pada gambar di bawah ini.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/indraazimi/mobpro1-new.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a `README`, `LICENSE`, and `.gitignore`.

...or create a new repository on the command line

```
echo "# mobpro1-new" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin https://github.com/indraazimi/mobpro1-new.git  
git push -u origin main
```

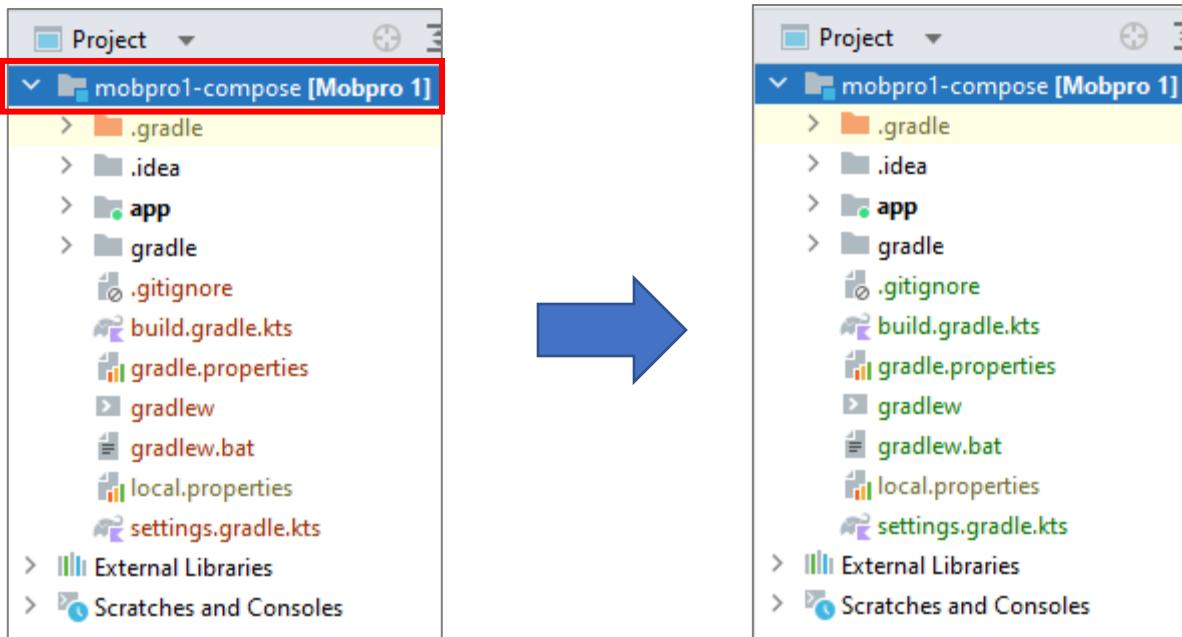
...or push an existing repository from the command line

```
git remote add origin https://github.com/indraazimi/mobpro1-new.git  
git branch -M main  
git push -u origin main
```

Copy perintah di atas, lalu pada Terminal Android Studio, paste baris tersebut, lalu tekan Enter. Jika tidak keluar pesan apa-apa, berarti sudah benar. Pastikan perintah yang di-copy/paste telah sesuai dengan repo masing-masing, karena tiap orang repository nya berbeda! Gambar di bawah ini hanya untuk ilustrasi. Tampilan Android Studio kamu mungkin berbeda.

```
\mobpro1-compose> git init  
/My Documents/Pemrograman/Mobpro2/mobpro1-compose/.git/  
\mobpro1-compose> git remote add origin https://github.com/indraazimi/mobpro1-compose.git  
\mobpro1-compose>
```

Langkah selanjutnya, lakukan penambahan file ke dalam stage yang di-tracked untuk versioning. Ubah mode Project Explorer dari Android ke Project. Setelah itu klik kanan pada nama project (yang ditandai merah pada gambar di halaman berikut), pilih menu Git > Add. Efek dari berhasilnya kode ini adalah, nama file akan berubah dari warna merah menjadi hijau.



Selanjutnya, pastikan setting nama dan email dari Git sudah benar. Caranya, ketik perintah berikut di Terminal Android Studio, lalu perhatikan outputnya.

```
Terminal: Local × + ▾
PS D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose> git config user.name
Indra Azimi
PS D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose> git config user.email
indraazimi@yahoo.com
PS D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose>
```

The screenshot shows the Android Studio terminal window with the following command history:

- `git config user.name` returns `Indra Azimi`
- `git config user.email` returns `indraazimi@yahoo.com`
- `PS D:\My Documents\Pemrograman\Mobpro2\mobpro1-compose>`

Below the terminal, the navigation bar includes tabs for Git, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, and Services.

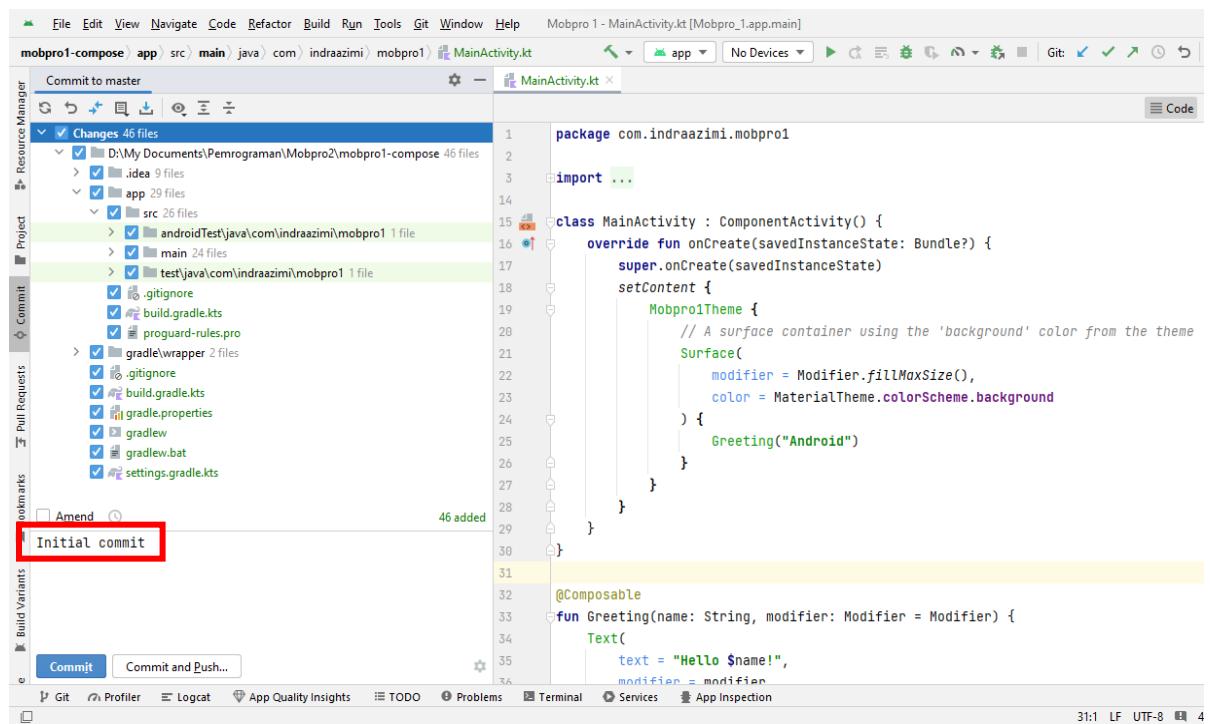
Pastikan nama telah menampilkan nama lengkap (*ingat, ini nama lengkap, bukan username Github*). Pastikan juga email telah sesuai dengan email yang digunakan di Github. Jika setelah dicek ternyata nama dan email masih belum sesuai, lakukan perubahan nama dan email dengan 2 perintah berikut:

Mengubah nama → `git config --global user.name "Nama Lengkap"`

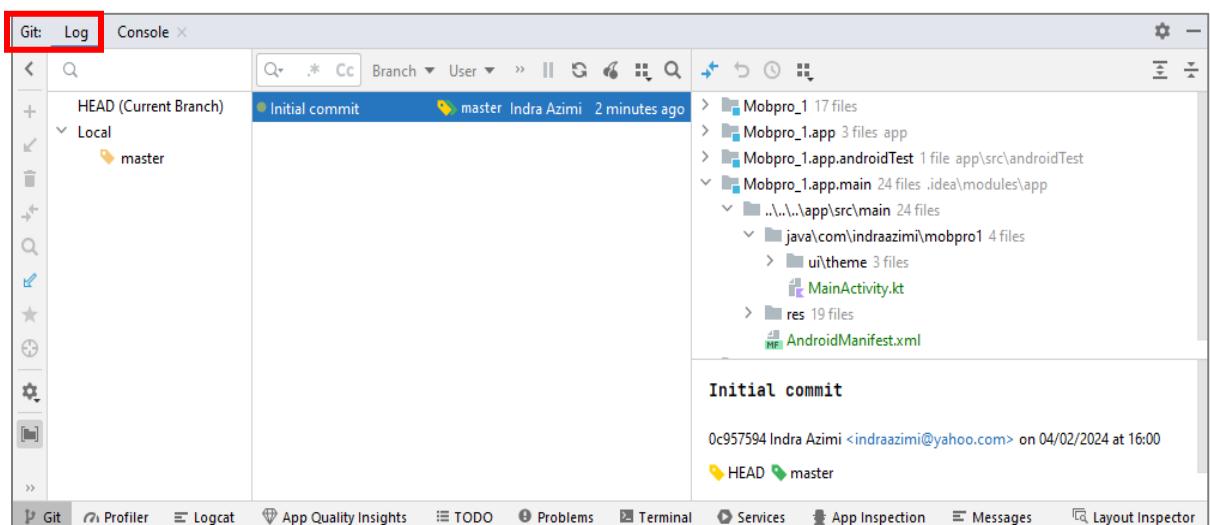
Mengubah email → `git config --global user.email "email@github.com"`

Jika informasi nama lengkap dan email sudah sesuai, lakukan commit pertama pada project ini. Caranya dengan memilih menu Git > Commit... atau bisa juga dengan langsung membuka tab Commit di pinggir kiri Android Studio, seperti terlihat pada gambar berikut.

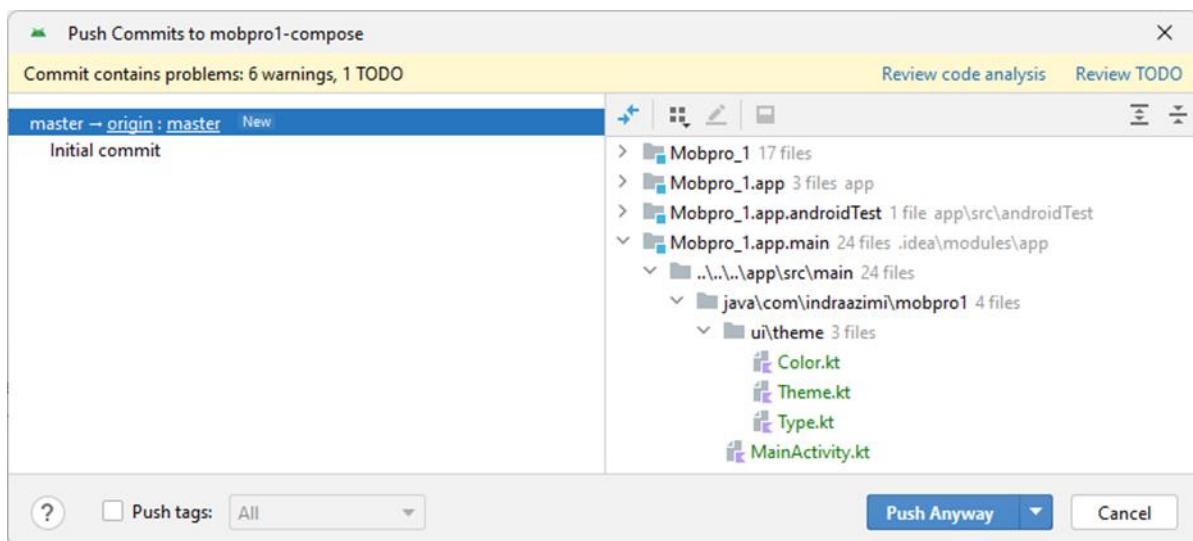
Isi commit message dengan “Initial commit”, lalu klik tombol Commit.



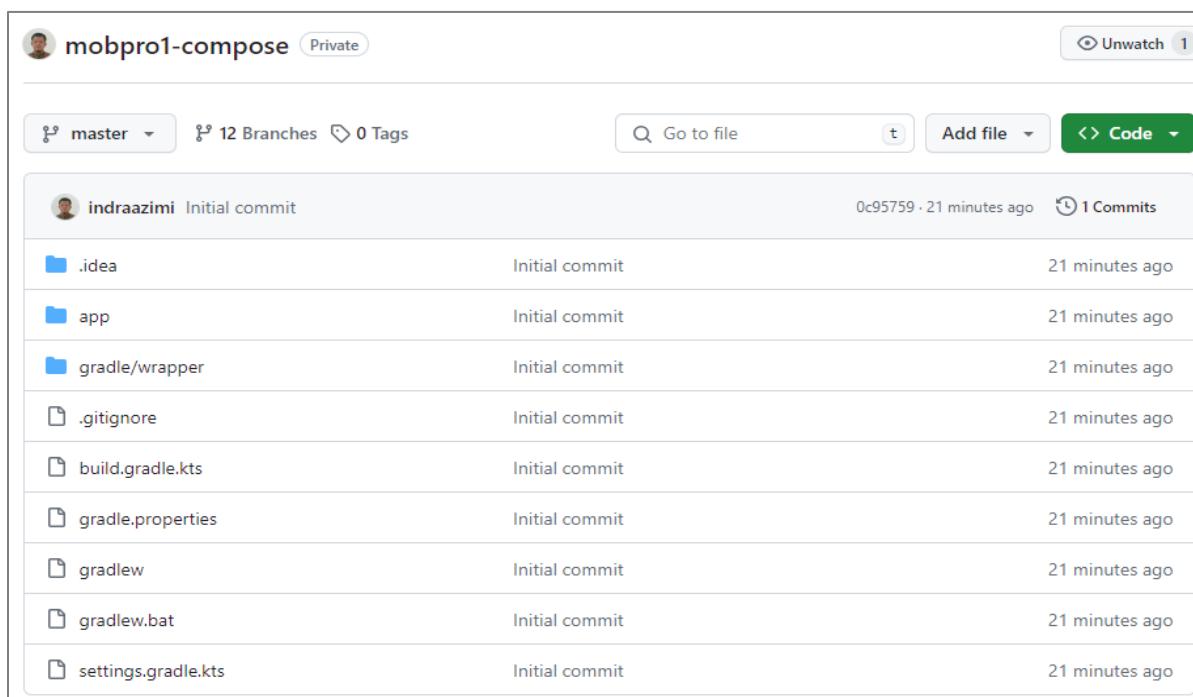
Efek dari berhasilnya langkah ini adalah, nama file pada Project Explorer akan berubah warna dari hijau menjadi hitam. Selain itu, jika kita cek di bagian Git > Log maka akan muncul pesan commit yang telah dibuat sebelumnya. Di bagian kanan bawah akan terlihat nama dan email yang melakukan commit, waktu commit, dan file-file yang ditambahkan di commit ini.



Berikutnya, kita akan melakukan operasi push ke repository Github. Caranya, klik menu Git > Push...
Pastikan initial commit telah tampil, lalu klik Push.



Pada dialog yang muncul selanjutnya, pilih akan login ke Github menggunakan username/password, atau menggunakan Personal Access Token (PAT). Cara mendapatkan PAT dari Github dapat mengikuti langkah-langkah yang terdapat pada link [ini](#). Push dapat dikatakan berhasil jika setelah di-refresh, repository Github yang awalnya kosong menjadi berisi folder dan file sesuai project dari Android Studio kita. Pastikan isinya lengkap seperti ini.



Berikutnya, kita akan menambahkan beberapa komponen pada project Android Studio ini. Kerjakan task-task pada modul ini di branch yang baru saja kita buat, langkah demi langkah.. Semoga berhasil!

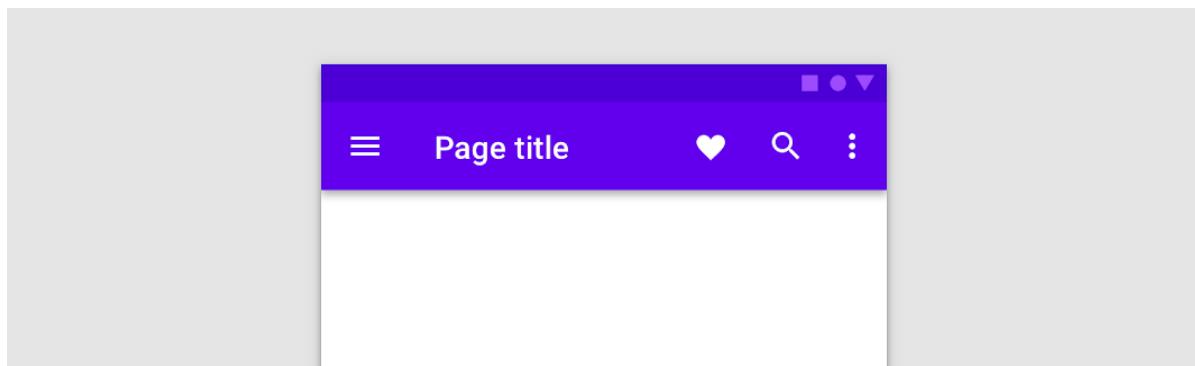
Dilarang keras untuk **copy – paste kode dari modul/sumber lain!**

**Ngoding pelan-pelan akan membuat kamu lebih jago di masa depan.
Lakukan commit setiap selesai 1 sub-task. Selamat ngoding!**

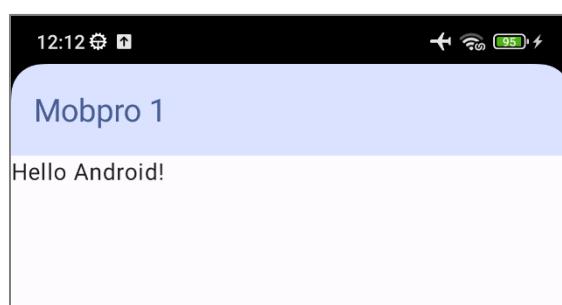
3. Task

3.1. Menambahkan Top App Bar

Setiap aplikasi membutuhkan identitas, yang akan membuat penggunanya tahu dan kemudian ingat mengenai aplikasi yang sedang mereka pakai. Komponen yang lazim digunakan untuk fungsi ini adalah Top App Bar. Selain digunakan untuk branding, komponen ini juga dapat digunakan untuk judul UI, berpindah dari satu tampilan ke tampilan lain, serta untuk menampilkan ikon aksi di aplikasi. Berikut contoh Top App Bar, diambil dari dokumentasi [Material Design](#).



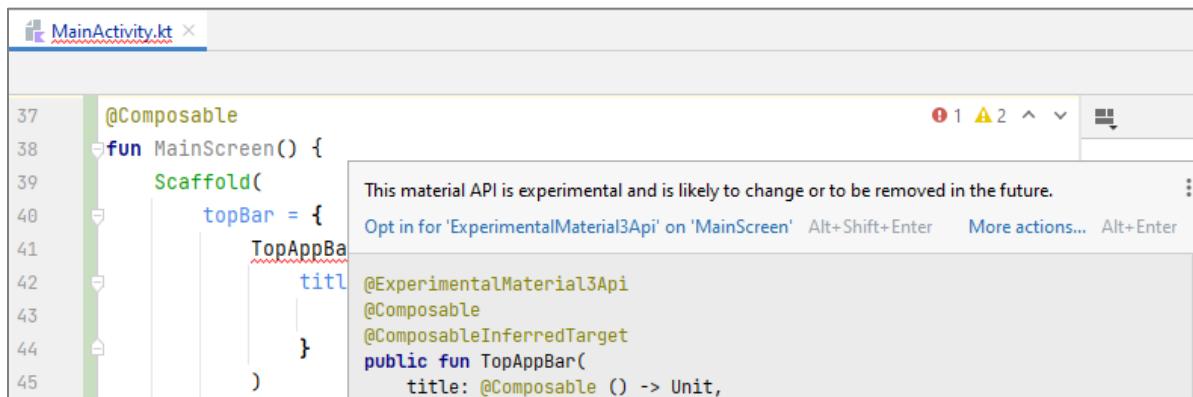
Karena sekarang masih awal, kita akan membuat App Bar yang sederhana, yang hanya memuat nama aplikasi kita saja. Berikut adalah tampilan setelah task ini selesai.



Buka file MainActivity, kemudian buat sebuah method baru bernama MainScreen. Letakkan method ini sebelum method Greeting.. Ketika mengetikkan source code, selalu gunakan auto-complete dari Android Studio. Jangan gunakan Android Studio seperti Notepad.

```
@Composable
fun MainScreen(content: @Composable (Modifier) -> Unit) {
    Scaffold(
        topBar = {
            TopAppBar(
                title = {
                    Text(text = stringResource(id = R.string.app_name))
                }
            )
        } { padding ->
            content(Modifier.padding(padding))
        }
}
```

Jika sudah, akan ada error di bagian TopAppBar. Arahkan kursor ke atasnya, maka akan muncul keterangan error seperti ini. Mengetahui error yang terjadi di kode kita sangatlah penting. Baca baris pertama **pesan error** ini, kita jadi tahu apa masalah kode kita.



Android Studio adalah editor yang pintar. Biasanya, di baris selanjutnya dari pesan error, kita akan menemukan solusi atas error tersebut. Klik teks “Opt-in for ...” maka akan ada kode baru yang ditambahkan di bagian atas method MainScreen, dan error tersebut menjadi hilang. Jika tidak, berarti masih ada error lain yang harus ditangani. Jangan panik.. 😊

Error lain yang sering terjadi adalah terdapat kelas yang belum di-import. Cara menanganinya kurang lebih sama. Arahkan kursor ke atas kode yang error, kemudian baca pesan errornya. Berdasarkan pesan error, pilih aksi sesuai yang disarankan Android Studio.

Selanjutnya kita akan menggunakan Top App Bar yang sudah kita buat ke dalam method Greeting.

Masih di MainActivity, cari kode Greeting dan ubah menjadi seperti ini:

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    fun Greeting(name: String) {
        MainScreen { modifier ->
            Text(
                text = "Hello $name!",
                modifier = modifier
            )
        }
    }
}
```

Kode berwarna abu adalah kode existing yang tidak berubah.

Kode yang dicoret adalah kode lama yang dihapus atau diubah.

Kode di bawahnya adalah kode baru yang harus ditambahkan.

Selanjutnya, pilih tampilan Split, akan terlihat preview dari aplikasi kita.



Dari tampilan preview, kita tahu bahwa App Bar kita belum ada warnanya. Untuk itu, edit kembali kode pada method MainScreen dan tambahkan kode berikut.

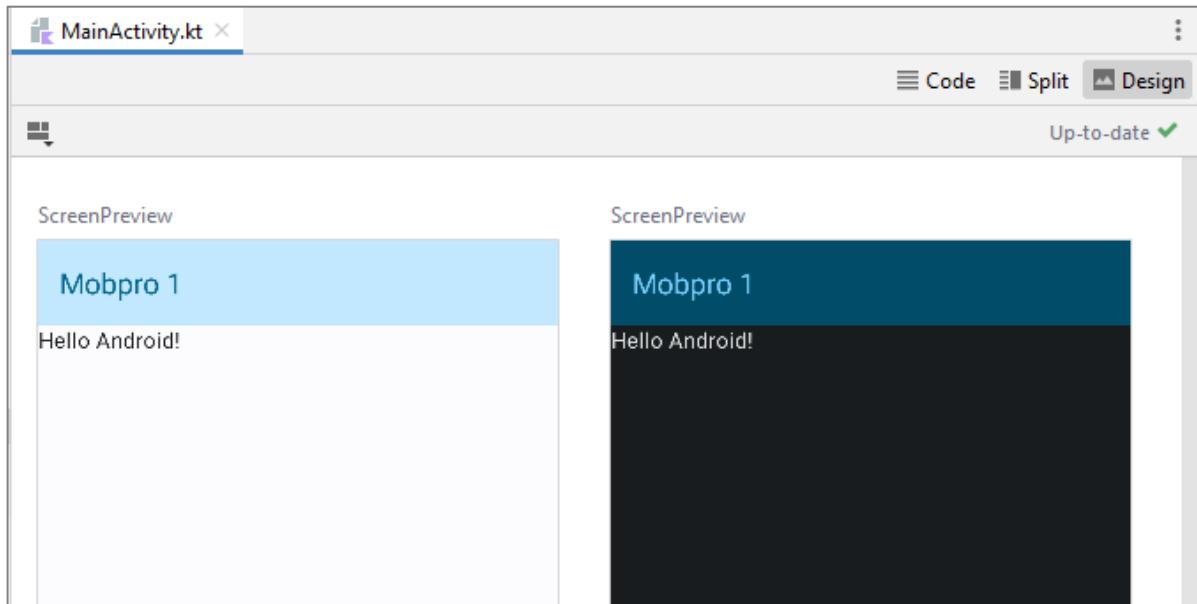
```
TopAppBar(
    title = {
        Text(text = stringResource(id = R.string.app_name))
    },
    colors = TopAppBarDefaults.mediumTopAppBarColors(
        containerColor = MaterialTheme.colorScheme.primaryContainer,
        titleContentColor = MaterialTheme.colorScheme.primary,
    )
)
```

Sekarang tampilan aplikasi sudah jadi lebih baik bukan? 😊

Jetpack Compose juga mendukung tema gelap. Untuk melihat tampilan aplikasi di tema gelap, tambahkan baris berikut di bagian preview. Sekalian ubah nama method agar sesuai kondisi terbaru.

```
@Preview(showBackground = true)
@Preview(uiMode = Configuration.UI_MODE_NIGHT_YES, showBackground = true)
@Composable
fun GreetingPreview() {
    ScreenPreview {
        Mobpro1Theme {
            Greeting("Android")
        }
    }
}
```

Kita juga bisa pindah tampilan ke mode Design seperti ini.



Sekarang jalankan aplikasi dengan menekan tombol Run. Apakah tampilan aplikasi di smartphone telah sama dengan tampilan preview di Android Studio?

Jika aplikasi telah berjalan sesuai harapan, silahkan lakukan commit dengan commit message:

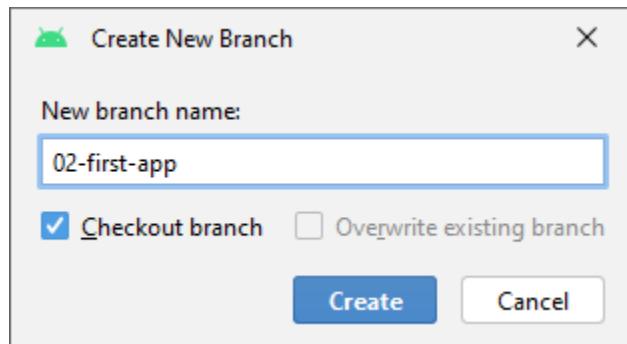
Menambahkan Top App Bar

Selanjutnya lakukan Git > Push.. Pastikan repo Github ter-update. Task 3.1 sudah selesai. Yey.. 🎉 🎉

Sebelum melanjutkan, ingat baik-baik bahwa ketika mengembangkan aplikasi, adalah wajar jika dalam prosesnya terjadi error, atau hal lain yang merintangi. Developer hebat bukanlah yang bisa ngoding tanpa error sama sekali. Tetapi, developer hebat adalah yang bisa mengatasi error ketika terjadi.. 😊

3.2. Membuat Tampilan Counter

Pada task ini, kita akan membuat tampilan counter. Akan tetapi, sebelum melakukan hal tersebut, klik dulu menu Git > New Branch. Masukkan “02-first-app” sebagai nama branch yang baru, lalu klik Create seperti terlihat pada gambar berikut.



Oleh karena tampilan kita tersusun secara vertikal, dalam artian ada Text, kemudian ada Button di bawahnya, maka kita harus menggunakan komponen Column untuk membungkus keduanya.. Ingat bahwa komponen Column ini harus diletakkan di dalam komponen MainScreen agar tampilan aplikasi mempunyai Top App Bar dari Task 3.1.

Buat method baru bernama Counter, kemudian tambahkan komponen MainScreen dan Column di dalamnya. Atur padding sebesar 16dp agar tampilan tidak kena pinggir layar. Atur juga agar isinya nanti berada di tengah-tengah layar.

```
@Composable
fun Counter() {
    MainScreen { modifier ->
        Column(
            modifier = modifier.fillMaxSize().padding(16.dp),
            verticalArrangement = Arrangement.Center,
            horizontalAlignment = Alignment.CenterHorizontally
        )
    }
}
```

Selanjutnya, tambahkan komponen Text di dalam komponen Column. Untuk saat ini, berikan teks default berupa angka “0” dan berikan juga style yang diperlukan agar teks yang muncul cukup besar sehingga mudah dilihat pengguna.

```
Column(  
    ...  
    horizontalAlignment = Alignment.CenterHorizontally  
) {  
    Text(  
        text = "0",  
        style = MaterialTheme.typography.displayLarge  
    )  
}
```

Panggil method Counter dan hapus method Greeting karena sudah tidak dipakai lagi.

```
setContent {  
    Mobpro1Theme {  
        Surface(  
            modifier = Modifier.fillMaxSize(),  
            color = MaterialTheme.colorScheme.background  
        ) {  
            Greeting("Android")  
            Counter()  
        }  
    }  
}  
  
@Composable  
fun ScreenPreview() {  
    Mobpro1Theme {  
        Greeting("Android")  
        Counter()  
    }  
}
```

Perhatikan tampilan preview. Pastikan sudah muncul angka 0 di tengah-tengah layar.

Selanjutnya, tambahkan komponen Button di bawah komponen Text. Atur lebar tombol sebesar setengah ukuran layar. Atur jarak tombol ke atas sebesar 16dp agar tidak menempel ke komponen Text di atasnya. Atur pula contentPadding agar tombol lebih besar dan mudah diklik.

```
Text(  
    text = index.toString(),  
    style = MaterialTheme.typography.displayLarge  
)  
Button(  
    onClick = {},  
    modifier = Modifier.fillMaxWidth(0.5f).padding(top = 16.dp),  
    contentPadding = PaddingValues(16.dp)  
) {  
}
```

Selanjutnya, buka file strings.xml yang terletak di folder res > values. Kita harus memasukkan setiap teks yang muncul ke pengguna ke dalam file ini. Tujuannya agar semua teks terkumpul di satu tempat sehingga aplikasi lebih mudah diterjemahkan.

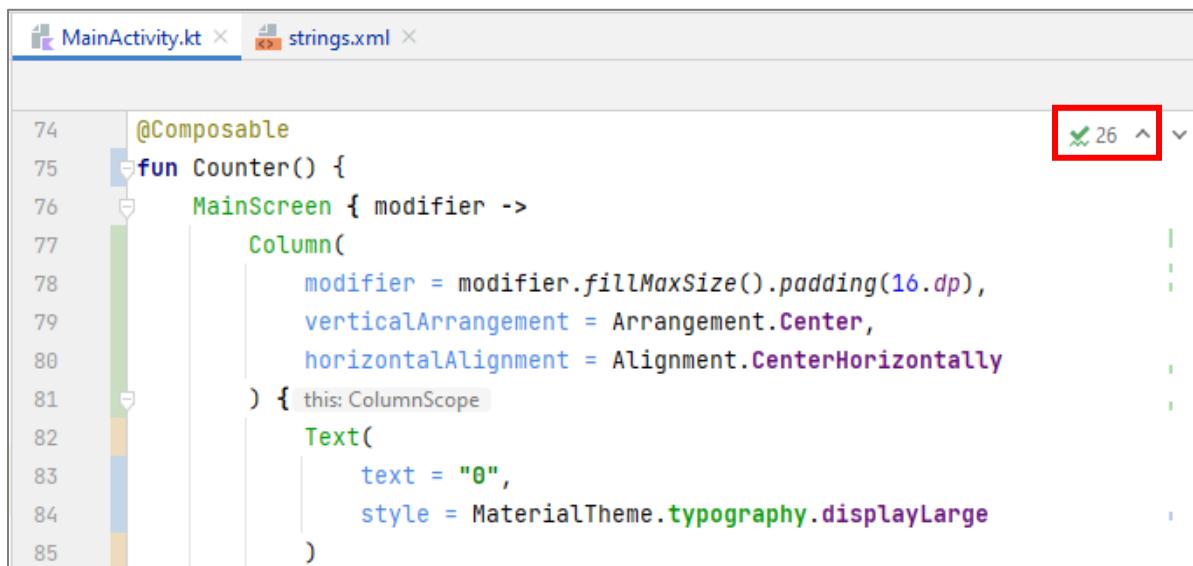
```
<resources>
    <string name="app_name">Mobpro 1</string>
    <string name="count">Count +1</string>
</resources>
```

Gunakan string tadi sebagai teks pada tombol.

```
Button(
    onClick = {},
    modifier = Modifier.fillMaxWidth(0.5f).padding(top = 16.dp),
    contentPadding = PaddingValues(16.dp)
) {
    Text(text = stringResource(R.string.count))
}
```

Sampai di sini, tampilan aplikasi sudah jadi. Jalankan dulu aplikasi di smartphone/emulator untuk memastikan hasil telah sesuai harapan.

Pastikan pula kode kita bersih yang ditandai adanya ceklis di kanan atas Code Editor. Jika masih ada tanda seru (warning) karena ada kode lama yang tidak dipakai dan belum dihapus, silahkan diperbaiki. Pastikan juga kode kalian telah rapi.



The screenshot shows the Android Studio code editor with two tabs: MainActivity.kt and strings.xml. The MainActivity.kt tab is active, displaying the following code:

```
74     @Composable
75     fun Counter() {
76         MainScreen { modifier ->
77             Column(
78                 modifier = modifier.fillMaxSize().padding(16.dp),
79                 verticalArrangement = Arrangement.Center,
80                 horizontalAlignment = Alignment.CenterHorizontally
81             ) { this: ColumnScope
82                 Text(
83                     text = "0",
84                     style = MaterialTheme.typography.displayLarge
85             )
```

A red box highlights a green checkmark icon with the number '26' next to it, located in the top right corner of the code editor window, indicating the count of warnings or errors.

Jika sudah, lakukan commit sesuai judul task.

3.3. Memberi Aksi Pada Tombol Saat Diklik

Task berikutnya adalah membuat logic aplikasi agar ketika Button diklik, angka yang muncul di Text otomatis bertambah satu. Untuk itu, kita membutuhkan sebuah variabel angka dengan jenis integer.. Tambahkan variable number di method Counter, kemudian lakukan import yang diperlukan.

```
@Composable
fun Counter() {
    var number by remember { mutableIntStateOf(0) }

    MainScreen { modifier ->
        ...
    }
}
```

Masih di method Counter, ubah komponen Text agar menampilkan angka tersebut.

```
Text(
    text = "0",
    text = number.toString(),
    style = MaterialTheme.typography.displayLarge
)
```

Berikutnya, buatlah agar ketika Button diklik, angka bertambah satu seperti kode ini.

```
Button(
    onClick = {},
    onClick = { number++ },
    modifier = Modifier.fillMaxWidth(0.5f).padding(top = 16.dp),
    contentPadding = PaddingValues(16.dp)
) {
    Text(text = stringResource(R.string.count))
}
```

Jalankan aplikasi.. Jika hasilnya sudah sesuai harapan, lakukan commit sesuai judul task.

3.4. Mendukung Bahasa Lokal (Indonesia)

Sadarkah kamu bahwa aplikasi kita saat ini menggunakan Bahasa Inggris? Agar aplikasi lebih mudah dimengerti, biasanya aplikasi akan diterjemahkan ke dalam bahasa negara tertentu. Proses ini disebut sebagai *localization*. Untungnya, Android mendukung proses lokalisasi dengan mudah. Syaratnya, semua teks yang muncul ke pengguna harus dimasukkan ke strings.xml. Oleh sebab itu, jangan pernah menulis string langsung di Kotlin alias di-hardcode.

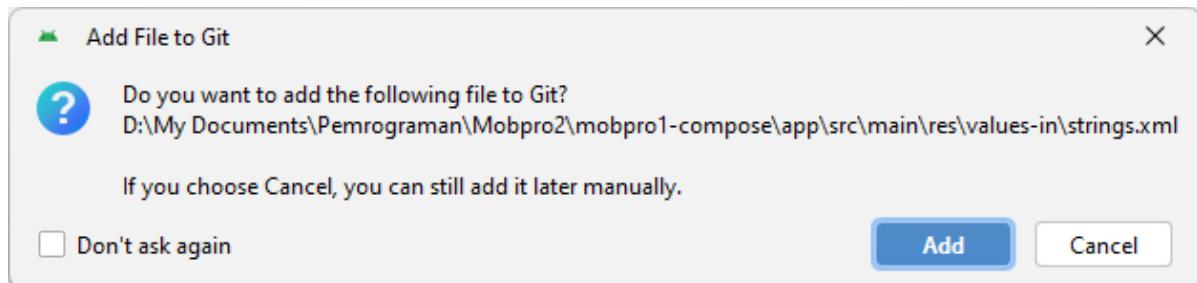
Buka file strings.xml kemudian klik Open Editor.



Setelah itu, klik ikon Add Locale dan pilih bahasa yang diinginkan, yaitu Bahasa Indonesia kita tercinta.

Translations Editor			
Key	Resource Folder	Untranslatable	Default Value
app_name	app/src/main/res	<input type="checkbox"/>	Mobpro 1
count	app/src/main/res	<input type="checkbox"/>	Count +1

Apabila muncul pertanyaan seperti ini, pilih Add agar string terjemahan bisa masuk ke dalam Git.



Kamu akan melihat ada 2 file strings.xml di Project Explorer: satu untuk teks default, dan satu lagi berisi teks Bahasa Indonesia.. Kembali ke Translations Editor, kemudian terjemahkan string untuk Bahasa Indonesia yang masih kosong dengan cara melakukan double click pada sel yang masih kosong.

The Project Explorer shows the following structure:

- values
 - colors.xml
 - strings (2)
 - strings.xml
 - strings.xml (in)
 - themes.xml

The Translations Editor table includes an additional column for the Indonesian locale:

Key	Resource Folder	Untranslatable	Default Value	Indonesian...
app_name	app/src/main/res	<input type="checkbox"/>	Mobpro 1	Mobpro 1
count	app/src/main/res	<input type="checkbox"/>	Count +1	Hitung +1

Jika sudah, jalankan kembali aplikasi di smartphone/emulator.

Jika perangkat kamu menggunakan Bahasa Indonesia, maka secara otomatis, aplikasi akan menggunakan Bahasa Indonesia juga. Keren bukan..? Jika perangkatmu menggunakan Bahasa Inggris, cobalah untuk mengubah setting bahasa perangkat ke Bahasa Indonesia agar hasil penerjemahan aplikasi dapat diverifikasi.

Jika semua sudah benar, maka selamat! Kamu telah berhasil menyelesaikan modul minggu ini dengan baik. Silahkan lakukan commit sesuai judul task, pastikan internet dalam keadaan aktif, kemudian lakukan push ke Github.

4. Summary

Pada modul ini kita telah membuat sebuah aplikasi Android menggunakan beberapa komponen yang umum dipakai, diantaranya: Top App Bar, Column, Text, dan juga Button. Kita juga telah melakukan pemberian aksi sederhana pada tombol yang ketika diklik akan berpengaruh pada data angka yang muncul di aplikasi. Selain itu, kita juga telah membuat aplikasi yang mendukung dua bahasa berbeda.

Selain membuat aplikasi, pada modul ini, kita juga menggunakan version control system berupa Git dan juga layanan dari Github. Terdapat operasi dasar berupa git init, git remote add origin, add file, commit, dan juga push untuk menyimpan kode versioning kita ke repository Github sehingga bisa diakses oleh siapapun yang memiliki hak akses.

5. Self-Reflection

Setiap kode yang kita tulis ada karena sebuah alasan.

Sudahkah kamu paham alasannya..?

Jika belum, jangan ragu untuk bertanya ke dosen kelasmu.

No.	Pertanyaan	Ya / Tidak
1	Saya dapat menggunakan Git dan Github di Android Studio	
2	Saya dapat membuat tampilan sederhana dengan Compose	
3	Saya dapat memberikan aksi ke tombol sesuai kebutuhan	
4	Saya dapat melakukan lokalisasi aplikasi ke bahasa tertentu	

6. Challenge

Buatlah branch baru dengan nama “02-challenge”. Selanjutnya, ubah project praktikum kalian agar menjadi seperti gambar kanan.

Aplikasi memiliki dua buah Button (-) dan (+), serta sebuah Text. Jika pengguna menekan tombol (+) angka akan bertambah, dan jika pengguna menekan tombol (-) angka akan berkurang.

Hint: Agar tampilan tersusun secara horizontal, maka kita harus menggunakan komponen Row, jadi bukan Column.

Mobpro 1



Spesifikasi tambahan (tidak wajib):

- Terdapat limit angka tidak bisa negatif
- Mengubah warna dan tema aplikasi menggunakan Material

Theme Builder di <https://m3.material.io/theme-builder>

Selamat mengerjakan 😊