MAIN

1. **main() Function:**
   * **WidgetsFlutterBinding.ensureInitialized();**: Memastikan bahwa binding untuk Flutter sudah diinisialisasi sebelum menjalankan aplikasi.
   * **SystemChrome.setPreferredOrientations(...)**: Mengatur preferensi orientasi perangkat, dalam contoh ini diatur agar hanya mendukung orientasi portrait (terutama untuk perangkat seluler).
   * **Logger.init(...)**: Inisialisasi logger untuk aplikasi. Jika aplikasi berada dalam mode release (**kReleaseMode**), maka logger diatur ke **LogMode.live** (mungkin untuk melacak log secara real-time), jika tidak, diatur ke **LogMode.debug**.
   * **runApp(MyApp());**: Menjalankan aplikasi dengan widget **MyApp**.
2. **MyApp Class (StatelessWidget):**
   * **GetMaterialApp**: Ini adalah widget yang menyediakan fitur-fitur dari paket **get** dan diintegrasikan dengan framework Flutter. Widget ini menyediakan fungsi seperti manajemen rute, penanganan status aplikasi, dan lainnya.
     + **debugShowCheckedModeBanner**: Menyembunyikan banner mode debug pada sudut kiri atas aplikasi.
     + **theme**: Tema yang digunakan dalam aplikasi.
     + **translations**: Pengaturan untuk lokalizasi aplikasi.
     + **locale**: Menentukan bahasa aplikasi berdasarkan locale perangkat.
     + **fallbackLocale**: Locale cadangan jika aplikasi tidak dapat menemukan terjemahan untuk bahasa utama.
     + **title**: Judul aplikasi.
     + **initialBinding**: Mengikat objek pada saat inisialisasi aplikasi (didefinisikan dalam **InitialBindings**).
     + **initialRoute**: Rute awal aplikasi (didefinisikan dalam **AppRoutes**).
     + **getPages**: Daftar halaman (rute) yang dapat diakses dalam aplikasi (didefinisikan dalam **AppRoutes.pages**).
     + **AppRoutes**: Kelas yang mungkin berisi definisi rute dan halaman aplikasi.

APP ROUTES - WELCOME

GetPage(

      name: welcomeScreen,

      page: () => WelcomeScreen(),

      bindings: [

        WelcomeBinding(),

      ],

Pada bagian ini, terdapat penggunaan kelas **GetPage** dari paket GetX. **GetPage** adalah objek konfigurasi yang digunakan untuk mendefinisikan rute pada aplikasi Flutter yang menggunakan GetX sebagai manajemen state dan navigasi.

1. **name: welcomeScreen,**
   * Ini adalah identifikasi unik untuk rute yang diberikan. Nantinya, digunakan saat mau pindah ke rute tertentu
2. **page: () => WelcomeScreen(),**
   * ketika rute **welcomeScreen** diakses, fungsi ini akan mengembalikan instance dari **WelcomeScreen()**. Dengan kata lain, ini adalah widget yang akan ditampilkan saat rute tersebut diakses.
3. **bindings: [WelcomeBinding(),],**
   * adalah daftar yang terkait dgn rute ini, digunakan untuk menginisialisasi objek atau melakukan tugas tertentu saat rute diakses. Misalnya setelah tampil halaman ini, saya ingin ke halaman onboarding. Ini termasuk implementasi dari controller dengan menggunakan getxcontroller.

**WelcomeController**

secara umum, **WelcomeController** ini digunakan untuk mengelola state dan mengarahkan navigasi dari layar sambutan (**WelcomeScreen**) ke rute berikutnya (**onboardingSatuScreen**).

* + **Rx<WelcomeModel> welcomeModelObj** adalah variabel Rx (reaktif) yang mengamati perubahan pada objek **WelcomeModel**. Dengan menggunakan Rx, saat nilainya berubah, widget yang menggunakannya akan diperbarui secara otomatis.

1. **onReady() Method:**
   * Metode ini dipanggil saat controller udah siap.
   * **Future.delayed** yang mengatur penundaan sebesar 3000 milidetik (3 detik).
   * Setelah penundaan, maka akan langsung pindah ke rute selanjutnya, nah inilah fungsinya **Get.toNamed** agar bisa pindah ke rute yg lain. Dalam hal ini, setelah tiga detik, pengguna akan diarahkan ke rute **onboardingSatuScreen** menggunakan **Get.toNamed(AppRoutes.onboardingSatuScreen)**.

**Penjelasan Code :**

1. **WelcomeScreen Class:**
   * Kelas ini merupakan widget yang menangani tampilan layar sambutan (welcome screen).
   * Extends **GetWidget<WelcomeController>** menunjukkan bahwa widget ini akan terkait dengan **WelcomeController** sebagai controller.
2. **build Method:**
   * **SafeArea** dan **Scaffold** untuk memastikan tampilan aman dan menambahkan struktur dasar halaman.
   * **CustomImageView** untuk menampilkan gambar
   * SizedBox untuk memerikan gap / spasi
3. **\_buildWelcomeSection Method:**
   * metode untuk membangun dan mengembalikan bagian tampilan yang berisi pesan selamat datang pada layar

APP ROUTES -