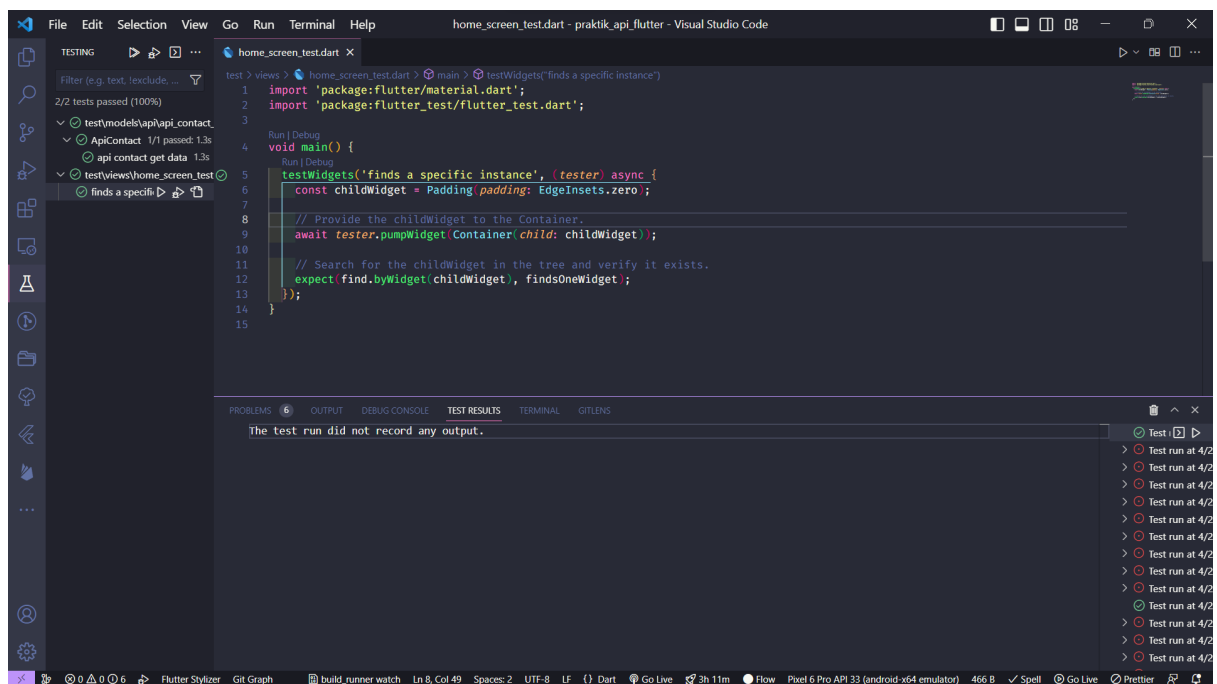Nama : Mohd Ryan Obillah
Kelas : SIB 4 Flutter Kelas B

Soal Prioritas 1 (80)

- Bukalah tugas (11. Soal Form & Advance Form Input *). Lakukan Test seluruh halaman yang pernah dibuat pada bagian judul, body, dan seluruh elemen yang ada pada halaman tersebut.
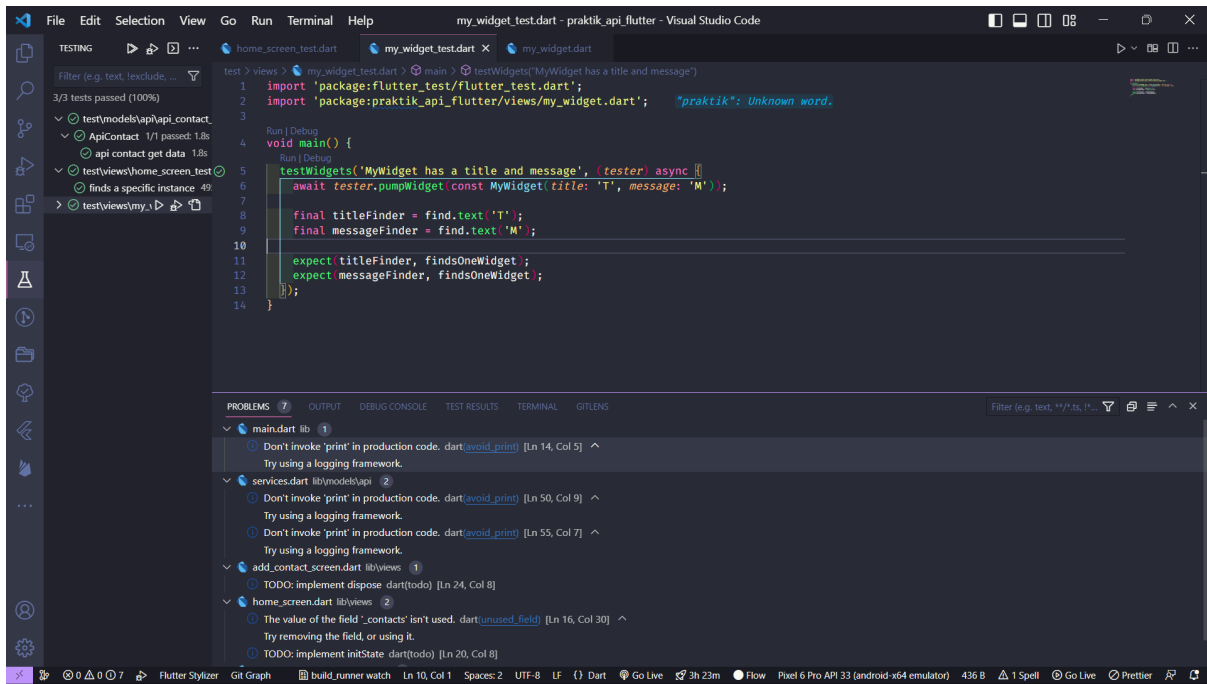
  Jawab :

Soal Prioritas 2 (20)

- Lakukan testing UI pada halaman yang belum pernah dibuat!

  Jawab :

Soal Eksplorasi (20)

1. Silahkan pilih aplikasi yang telah dikembangkan.
2. Lakukan UI Testing pada aplikasi yang telah dikembangkan. Pastikan semua fitur dilakukan UI Testing.

First window — login_screen_test.dart

```dart
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:flutter_test/flutter_test.dart';
import 'package:lapkin_mobile/views/screens/login_screen.dart';

void main() {
  testWidgets('Login Screen UI test', (tester) async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();
    // Build MyPage widget
    await tester.pumpWidget(const ProviderScope(
      child: MediaQuery(
        data: MediaQueryData(),
        child: MaterialApp(
          home: LoginScreen(),
        )), // MaterialApp // MediaQuery
    )); // ProviderScope

    // Verify the UI widgets
    expect(find.text('Selamat Datang di LAPKIN'), findsOneWidget);
    expect(find.text('Silahkan Login Untuk Memulai'), findsOneWidget);
    expect(find.byType(Image), findsOneWidget);
  });
}
```

Second window — login_form_widget_test.dart

```dart
import 'package:flutter/material.dart';
import 'package:flutter_test/flutter_test.dart';
import 'package:lapkin_mobile/views/widgets/login_form_widget.dart';

void main() {
  testWidgets('login form widget test', (tester) async {
    await tester.pumpWidget(const MediaQuery(
      data: MediaQueryData(),
      child: MaterialApp(
        home:
      )); //
    expect(find
    expect(find
  });
}
```

Autocomplete popup:
```
(const) MaterialApp MaterialApp({
  Key? key,
  GlobalKey<NavigatorState>? navigatorKey,
  GlobalKey<ScaffoldMessengerState>? scaffoldMessengerKey,
  Widget? home,
  Map<String, Widget Function(BuildContext)> routes = const <String, WidgetBuilder>{},
  String? initialRoute,
  Route<dynamic>? Function(RouteSettings)? onGenerateRoute,
  List<Route<dynamic>> Function(String)? onGenerateInitialRoutes,
  Route<dynamic>? Function(RouteSettings)? onUnknownRoute,
  List<NavigatorObserver> navigatorObservers = const <NavigatorObserver>[],
  Widget Function(BuildContext, Widget?)? builder,
  String title = '',
  String Function(BuildContext)? onGenerateTitle,
  Color? color
```