

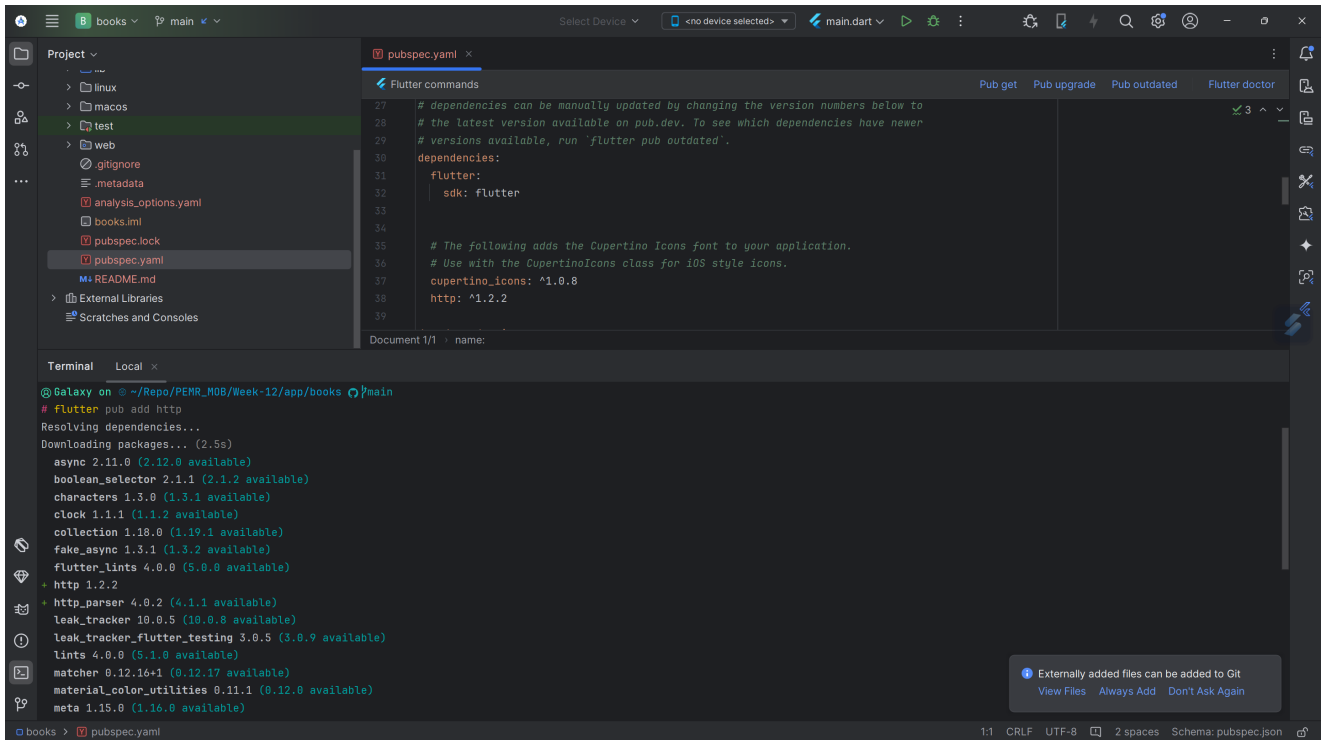
12 | Pemrograman Asynchronous

Praktikum 1: Mengunduh Data dari Web Service (API)

Langkah 1: Buat Project Baru

Buatlah sebuah project flutter baru dengan nama **books** di folder src **week-12** repository GitHub Anda.

Kemudian Tambahkan dependensi http dengan mengetik perintah **flutter pub get http**



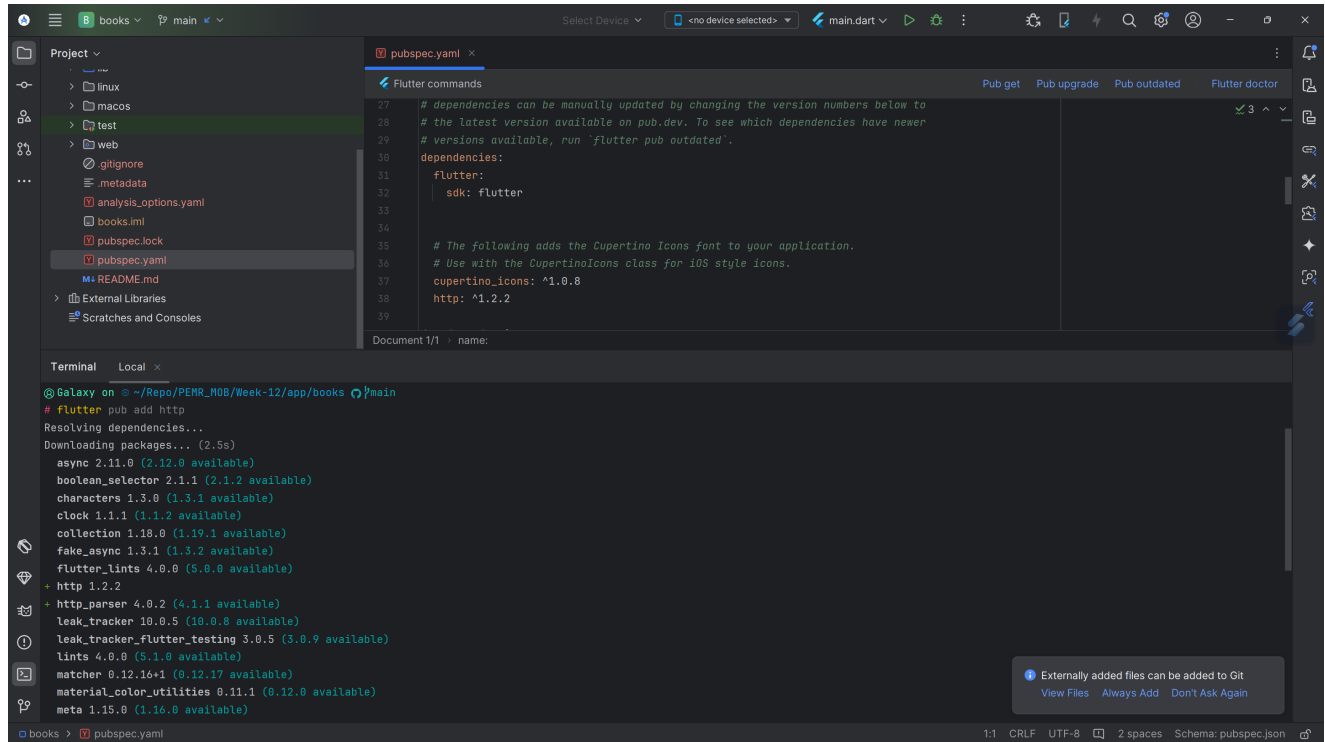
```
pubspec.yaml
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run 'flutter pub outdated'.
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  http: ^1.2.2

Terminal
@Galaxy on ~ /Repo/PEMR_MOB/Week-12/app/books main
# flutter pub add http
Resolving dependencies...
Downloading packages... (2.5s)
  async 2.11.0 (2.12.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.3.1 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.18.0 (1.19.1 available)
  fake_async 1.3.1 (1.3.2 available)
  flutter_lints 4.0.0 (5.0.0 available)
  http 1.2.2
  http_parser 4.0.2 (4.1.1 available)
  leak_tracker 10.0.5 (10.0.8 available)
  leak_tracker_flutter_testing 3.0.5 (3.0.9 available)
  lints 4.0.0 (5.1.0 available)
  matcher 0.12.16+1 (0.12.17 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
```

Langkah 2: Cek file pubspec.yaml

Jika berhasil install plugin, pastikan plugin http telah ada di file pubspec.



Langkah 3: Buka file main.dart

Ketiklah kode seperti berikut ini.

Soal 1

Tambahkan nama panggilan Anda pada title app sebagai identitas hasil pekerjaan Anda.

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:http/http.dart';
import 'package:http/http.dart' as http;

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Naufal',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: ,
    );
  }
}
```

```

}

class FuturePage extends StatefulWidget {
  const FuturePage({super.key});

  @override
  State<FuturePage> createState() => _FuturePageState();
}

class _FuturePageState extends State<FuturePage> {
  String result = '';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Back from the future'),
      ),
      body: Center(
        child: Column(children: [
          const Spacer(),
          ElevatedButton(
            onPressed: () {

              }, child: const Text('GO!')),
          const Spacer(),
          Text(result),
          const Spacer(),
          const CircularProgressIndicator(),
          const Spacer()
        ],),
      ),
    );
  }
}

```

Langkah 4: Tambah method getData()

Tambahkan method berikut ke dalam class `_FuturePageState` yang berguna untuk mengambil data dari *API Google Books*.

```

Future<Response> getData() async {
  const authority = 'www.googleapis.com';
  const path = '/books/v1/volumes/WBR2DwAAQBAJ';
  Uri url = Uri.https(authority, path);
  return http.get(url);
}

```

Carilah judul buku favorit Anda di Google Books, lalu ganti ID buku pada variabel path di kode tersebut.

```

{
  "kind": "books#volume",
  "id": "WBR2DwAAQBAJ",
  "etag": "1lucKcmPVs",
  "selfLink": "https://www.googleapis.com/books/v1/volumes/WBR2DwAAQBAJ",
  "volumeInfo": {
    "title": "Pemrograman C++ : Dasar Pemrograman Berorientasi Objek",
    "authors": [
      "Risonan Hasiholan Sianipar"
    ],
    "publisher": "Penerbit ANDI",
    "publishedDate": "2012-11-26",
    "description": "u003cp/u003ebuku ini dikhususkan bagi pembaca yang benar-benar ingin menguasai fondasi pemrograman berbasis objek. Karena fondasi harus kokoh, buku ini sungguh-sungguh memperdalam konsep-konsep yang mendasari PBO misalnya komposisi terpisah, namespace dan memori statis/dinamis (terutama tentang skop dan koneksi suatu variabel), overloading fungsi, dan kelas. Selain antara pointer, array, dan objek juga disajikan. Overloading operator dan aplikasinya juga didiskusikan secara kritis di dalam buku ini. Setiap kode sumber ditelaah sepotong demi sepotong agar memberikan penjelasan yang integratif kepada pembaca. u003cp/u003eu003cp/u003eu003cp/u003eu003cp/u003eu003cp/u003eSebelumnya beberapa topik konseptual yang disajikan dalam buku ini. u003cp/u003eu003cp/u003eu003cp/u003ekonsep penanganan tipe data dasar: bool, char, short, int, long, float, dan double. u003cp/u003eu003cp/u003eu003cp/u003ekonsep penanganan tipe data gabungan, array, string, struktur, union, enumerasi, dan pointer. u003cp/u003eu003cp/u003eu003cp/u003ekonsep penggunaan dan aplikasi operator new dan delete. u003cp/u003eu003cp/u003eu003cp/u003eloop dan ekspresi relasional. u003cp/u003eu003cp/u003eu003cp/u003ePercabangan dan operator logika. u003cp/u003eu003cp/u003eu003cp/u003ekonsep fungsi: relasi fungsi dan struktur, relasi fungsi dan array, relasi fungsi dan pointer, overloading fungsi dan lain-lain. u003cp/u003eu003cp/u003eu003cp/u003emodel memori dan namespace: Durasi, skop, dan koneksi suatu variabel. u003cp/p/u003eu003cp/u003eu003epenggunaan dan objek: Fungsi friend, konstruktor, destruktur, overloading operator, fungsi konversi kelas, dan lain-lain. u003cp/u003eu003cp/u003eu003e",
    "industryIdentifiers": [
      {
        "type": "ISBN_10",
        "identifier": "97929729211848"
      },
      {
        "type": "ISBN_13",
        "identifier": "9797929211846"
      }
    ],
    "readingModes": {
      "text": false,
      "image": true
    },
    "pageCount": 432,
    "printedPageCount": 407,
    "printType": "BOOK",
    "maturityRating": "NOT_MATURE",
    "allowAnonLogging": false,
    "contentVersion": "1.1.1.0.preview.1",
    "panelizationSummary": {
      "containsPubSubbles": false,
      "containsImageSubbles": false
    },
    "imageLinks": {
      "smallThumbnail": "http://books.google.com/books/publisher/content?id=WBR2DwAAQBAJ&printsec=frontcover&img=1&zoom=5&edge=curL&imgt&f=AFLE7yioXbo07p-6obJzJH6532jg_rIrt9J2XOHx108&oeq&4sU8p1phZue1Dm1KUp7&4Z3D&7&wqYU1REDC&U19U541fgr&sd&ga&wh&4jJ9&H9D&H6M.ch&v&wPGXND&HPC&M&source=gb&api",
      "thumbnail": "http://books.google.com/books/publisher/content?id=WBR2DwAAQBAJ&printsec=frontcover&img=1&zoom=1&edge=curL&imgt&f=AFLE7yioXbo07p-6obJzJH6532jg_rIrt9J2XOHx108&oeq&4sU8p1phZue1Dm1KUp7&4Z3D&7&wqYU1REDC&U19U541fgr&sd&ga&wh&4jJ9&H9D&H6M.ch&v&wPGXND&HPC&M&source=gb&api",
      "C&B&T&QV7&9B1IT&IK&f&N&f&AN&V&E16&S&M&C&AS&Y&N&f&w&My&DSK_y&w&f&C&O&V&ID&K&G1-&ol&Q&DY&V&4&w&R&M&T&I&source=gb&api",
      "medium": "http://books.google.com/books/publisher/content?id=WBR2DwAAQBAJ&printsec=frontcover&img=1&zoom=3&edge=curL&imgt&f=AFLE7yioXbo07p-6obJzJH6532jg_rIrt9J2XOHx108&oeq&4sU8p1phZue1Dm1KUp7&4Z3D&7&wqYU1REDC&U19U541fgr&sd&ga&wh&4jJ9&H9D&H6M.ch&v&wPGXND&HPC&M&source=gb&api"
    }
  }
}

```

Praktikum 2: Menggunakan await/async untuk menghindari callbacks

Praktikum 3: Menggunakan Completer di Future

Praktikum 4: Memanggil Future secara paralel

Praktikum 5: Menangani Respon Error pada Async Code

Praktikum 6: Menggunakan Future dengan StatefulWidget

Praktikum 7: Manajemen Future dengan FutureBuilder

Praktikum 8: Navigation route dengan Future Function

Praktikum 9: Memanfaatkan async/await dengan Widget Dialog