



UNGUIDED 12



Disusun Oleh :
Nama : Ganes Gemi Putra
Kelas : SE-07-02
NIM : (2311104075)

Dosen : YUDHA ISLAMI SULISTYA S.Kom M.Kom

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2025

1. Tujuan Praktikum

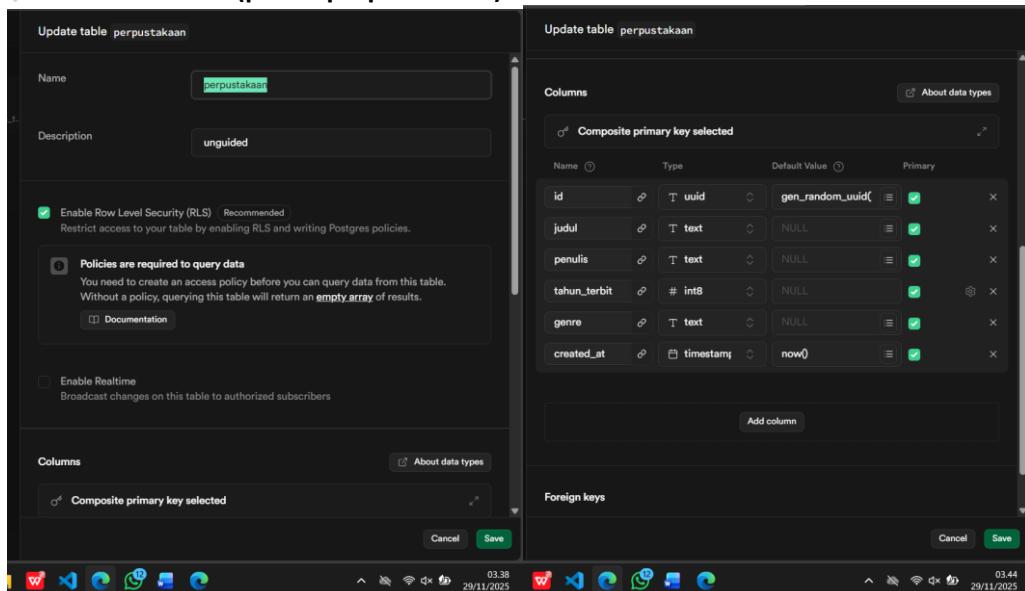
Praktikum ini bertujuan untuk:

1. Membuat database cloud menggunakan **Supabase**.
2. Membuat tabel **perpustakaan** untuk menyimpan biodata buku.
3. Menghubungkan aplikasi **Flutter** ke Supabase.
4. Melakukan operasi:
 - o **Create** (menambahkan data buku)
 - o **Read** (mengambil & menampilkan data)
5. Menampilkan hasil data dalam **terminal/console** & menampilkan total buku di UI.

2. Pembuatan Database di Supabase

Pada Supabase, dibuat sebuah project → lalu dibuat tabel **perpustakaan** dengan struktur:

❖ Struktur Tabel (public.perpustakaan)



Kolom	Tipe Data	Keterangan	Default
id	uuid	Primary Key	gen_random_uuid()
judul	text	-	-
penulis	text	-	-
tahun_terbit	integer	-	-
genre	text	-	-
created_at	timestamp	otomatis	now()

❖ Row Level Security (RLS)

Pada tabel terlihat **RLS aktif**, artinya harus memakai policy agar data bisa dibaca/ditulis. Supabase Flutter otomatis memakai anonKey → sudah bisa akses public data.

3. Implementasi di Flutter

Aplikasi Flutter menggunakan package:

```
supabase_flutter: ^2.0.0
```

Pada main.dart dilakukan:

❖ 1. Inisialisasi Supabase

```
await Supabase.initialize(
  url: 'https://rnqnokzhtskoydgzdybr.supabase.co',
  anonKey: '...'
);
```

❖ 2. Pembuatan Model Book

Model berfungsi memetakan data dari tabel Supabase:

```
class Book {  
    final String id;  
    final String judul;  
    final String penulis;  
    final int tahunTerbit;  
    final String genre;  
    final DateTime createdAt;  
  
    ...  
}
```

❖ 3. Service CRUD

Hanya digunakan 2 operasi:

Create (Insert)

```
_supabase.from('perpustakaan').insert(book.toJson());
```

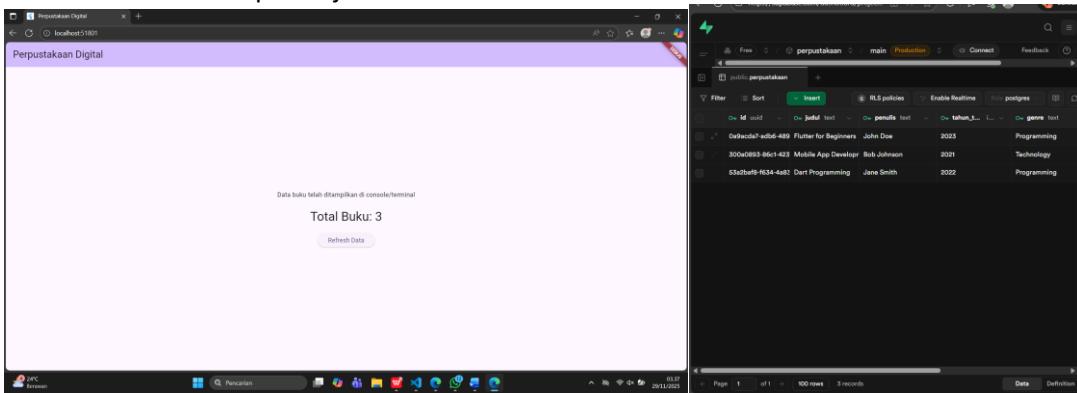
Read (Select)

```
final response = await _supabase  
    .from('perpustakaan')  
    .select('*')  
    .order('created_at', ascending: false);
```

4. Cara Kerja Aplikasi

Ketika aplikasi berjalan:

1. **Menambahkan 3 buku contoh (Create)**
 - ✓ Data langsung tersimpan di Supabase
 - ✓ Ditampilkan per buku di terminal
2. **Mengambil seluruh data dari tabel (Read)**
 - ✓ Data muncul di terminal
 - ✓ UI menampilkan jumlah total buku



ID	Judul	Penulis	Tahun Terbit	Genre
0xdeadcafe-4e00-4800-8000-000000000000	Flutter for Beginners	John Doe	2023	Programming
300e0893-86c1-4232-9000-000000000000	Mobile App Developer	Bob Johnson	2021	Technology
53d2baef-9344-4a87-9000-000000000000	Dart Programming	Jane Smith	2022	Programming



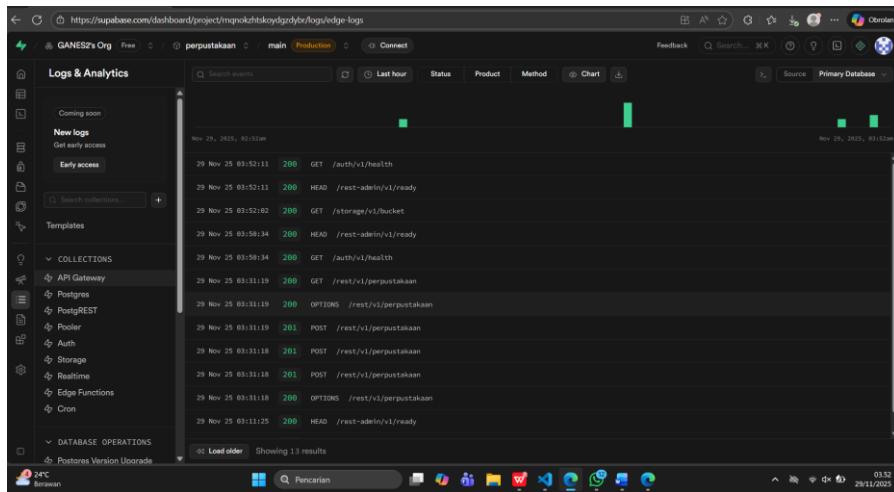
Pada terminal muncul:



```

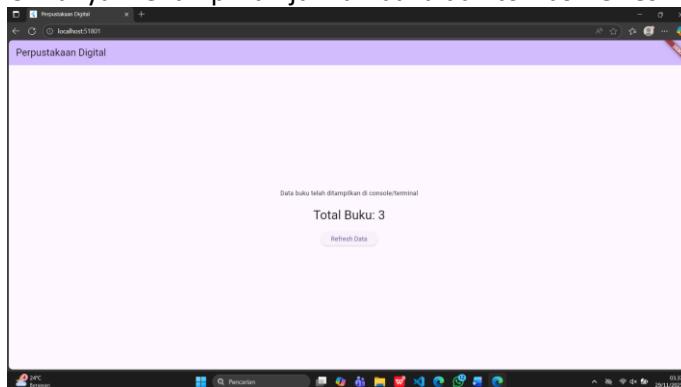
File Edit Selection View Go Run Terminal Help ↵
    Q unguished

EXPLORER   main.dart U
    lib > main.dart
        1: import 'package:flutter/material.dart';
        2: import 'package:supabase_flutter/supabase_flutter.dart';
        3:
        4: Run/Debug Profile
        5: void main() async {
        6:     WidgetsFlutterBinding.ensureInitialized();
        7:     await supabase.initializeApp(
        8:         url: "https://supabase-489f.firebaseio.com",
        9:         anonKey: "key"
        10:    );
        11:
        12:    runApp(const MyApp());
        13:
        14}
        15:
        16: class Book {
        17:     final String id;
        18:     final String judul;
        19:     final String penulis;
        20:     final int tahunTerbit;
        21:     final String genre;
        22:     final DateTime createdAt;
        23:
        24:     Book({
        25:         required this.id,
        26:         required this.judul,
        27:         required this.penulis,
        28:         required this.tahunTerbit,
        29:         required this.genre,
        30:         required this.createdAt,
        31:     });
        32:
        33:     factory Book.fromJson(Map<String, dynamic> json) {
        34:         return Book(
        35:             id: json['id'],
        36:             judul: json['judul'],
        37:             penulis: json['penulis'],
        38:             tahunTerbit: json['tahunTerbit'],
        39:             genre: json['genre'],
        40:             createdAt: json['createdAt'].toDate(),
        41:         );
        42:     }
        43: }
        44:
        45: Future<Book> fetchBook(String id) {
        46:     final String judul;
        47:     final String penulis;
        48:     final int tahunTerbit;
        49:     final String genre;
        50:     final DateTime createdDate;
        51:
        52:     Book book;
        53:
        54:     book = Book(
        55:         id: id,
        56:         judul: judul,
        57:         penulis: penulis,
        58:         tahunTerbit: tahunTerbit,
        59:         genre: genre,
        60:         createdDate: createdDate);
        61:
        62:     return book;
        63: }
        64:
        65: Future<List<Book>> fetchAllBooks() {
        66:     final String url =
        67:         "https://supabase-489f.firebaseio.com/books.json";
        68:     final response = await http.get(Uri.parse(url));
        69:     final List<Map> data = json.decode(response.body);
        70:     final List<Book> books = data.map((item) {
        71:         return Book.fromJson(item);
        72:     }).toList();
        73:
        74:     return books;
        75: }
        76:
        77: void main() {
        78:     runApp(MyApp());
        79: }
        80:
        81: class MyApp extends StatelessWidget {
        82:     @override
        83:     Widget build(BuildContext context) {
        84:         return MaterialApp(
        85:             title: 'Flutter Demo',
        86:             theme: ThemeData(
        87:                 primarySwatch: Colors.blue,
        88:             ),
        89:             home: MyHomePage(),
        90:         );
        91:     }
        92: }
        93:
        94: class MyHomePage extends StatefulWidget {
        95:     @override
        96:     _MyHomePageState createState() => _MyHomePageState();
        97: }
        98:
        99: class _MyHomePageState extends State<MyHomePage> {
        100:     @override
        101:     void initState() {
        102:         super.initState();
        103:     }
        104:
        105:     void _showSnackBar(String message) {
        106:         Scaffold.of(context).showSnackBar(SnackBar(
        107:             content: Text(message),
        108:         ));
        109:     }
        110:
        111:     void _fetchBooks() {
        112:         Future<List<Book>> books = fetchAllBooks();
        113:         books.then((value) {
        114:             setState(() {
        115:                 _books = value;
        116:             });
        117:         });
        118:     }
        119:
        120:     void _createBook() {
        121:         final String judul = _judulController.text;
        122:         final String penulis = _penulisController.text;
        123:         final int tahunTerbit = int.parse(_tahunTerbitController.text);
        124:         final String genre = _genreController.text;
        125:
        126:         Book book = Book(
        127:             id: null,
        128:             judul: judul,
        129:             penulis: penulis,
        130:             tahunTerbit: tahunTerbit,
        131:             genre: genre);
        132:
        133:         final String url =
        134:             "https://supabase-489f.firebaseio.com/books.json";
        135:         final response = await http.post(Uri.parse(url),
        136:             body: json.encode(book));
        137:         final Map data = json.decode(response.body);
        138:         final String id = data['id'];
        139:
        140:         setState(() {
        141:             _books.add(book);
        142:             _judulController.clear();
        143:             _penulisController.clear();
        144:             _tahunTerbitController.clear();
        145:             _genreController.clear();
        146:         });
        147:     }
        148:
        149:     void _updateBook(int index) {
        150:         final String id = _books[index].id;
        151:         final String judul = _judulController.text;
        152:         final String penulis = _penulisController.text;
        153:         final int tahunTerbit = int.parse(_tahunTerbitController.text);
        154:         final String genre = _genreController.text;
        155:
        156:         Book book = Book(
        157:             id: id,
        158:             judul: judul,
        159:             penulis: penulis,
        160:             tahunTerbit: tahunTerbit,
        161:             genre: genre);
        162:
        163:         final String url =
        164:             "https://supabase-489f.firebaseio.com/books/$id.json";
        165:         final response = await http.put(Uri.parse(url),
        166:             body: json.encode(book));
        167:         final Map data = json.decode(response.body);
        168:
        169:         setState(() {
        170:             _books[index] = book;
        171:             _judulController.clear();
        172:             _penulisController.clear();
        173:             _tahunTerbitController.clear();
        174:             _genreController.clear();
        175:         });
        176:     }
        177:
        178:     void _deleteBook(int index) {
        179:         final String id = _books[index].id;
        180:
        181:         final String url =
        182:             "https://supabase-489f.firebaseio.com/books/$id.json";
        183:         final response = await http.delete(Uri.parse(url));
        184:
        185:         setState(() {
        186:             _books.removeAt(index);
        187:         });
        188:     }
        189:
        190:     void _sortBooks(String sortType) {
        191:         if (sortType == 'Judul') {
        192:             _books.sort((a, b) {
        193:                 if (a.judul.toLowerCase() < b.judul.toLowerCase()) {
        194:                     return -1;
        195:                 } else if (a.judul.toLowerCase() > b.judul.toLowerCase()) {
        196:                     return 1;
        197:                 } else {
        198:                     return 0;
        199:                 }
        200:             });
        201:         } else if (sortType == 'Penulis') {
        202:             _books.sort((a, b) {
        203:                 if (a.penulis.toLowerCase() < b.penulis.toLowerCase()) {
        204:                     return -1;
        205:                 } else if (a.penulis.toLowerCase() > b.penulis.toLowerCase()) {
        206:                     return 1;
        207:                 } else {
        208:                     return 0;
        209:                 }
        210:             });
        211:         } else if (sortType == 'Tahun Terbit') {
        212:             _books.sort((a, b) {
        213:                 if (a.tahunTerbit < b.tahunTerbit) {
        214:                     return -1;
        215:                 } else if (a.tahunTerbit > b.tahunTerbit) {
        216:                     return 1;
        217:                 } else {
        218:                     return 0;
        219:                 }
        220:             });
        221:         } else if (sortType == 'Genre') {
        222:             _books.sort((a, b) {
        223:                 if (a.genre.toLowerCase() < b.genre.toLowerCase()) {
        224:                     return -1;
        225:                 } else if (a.genre.toLowerCase() > b.genre.toLowerCase()) {
        226:                     return 1;
        227:                 } else {
        228:                     return 0;
        229:                 }
        230:             });
        231:         }
        232:     }
        233:
        234:     void _refreshBooks() {
        235:         _fetchBooks();
        236:     }
        237:
        238:     void _refreshForm() {
        239:         _judulController.clear();
        240:         _penulisController.clear();
        241:         _tahunTerbitController.clear();
        242:         _genreController.clear();
        243:     }
        244:
        245:     void _refreshList() {
        246:         _refreshBooks();
        247:         _refreshForm();
        248:     }
        249:
        250:     void _refreshSort() {
        251:         _refreshList();
        252:         _sortBooks(sortType);
        253:     }
        254:
        255:     void _refreshDelete() {
        256:         _refreshList();
        257:     }
        258:
        259:     void _refreshUpdate() {
        260:         _refreshList();
        261:         _refreshForm();
        262:     }
        263:
        264:     void _refreshCreate() {
        265:         _refreshList();
        266:         _refreshForm();
        267:     }
        268:
        269:     void _refreshSearch() {
        270:         _refreshList();
        271:     }
        272:
        273:     void _refreshFilter() {
        274:         _refreshList();
        275:     }
        276:
        277:     void _refreshPage() {
        278:         _refreshList();
        279:     }
        280:
        281:     void _refreshSortType() {
        282:         _refreshList();
        283:         _sortBooks(sortType);
        284:     }
        285:
        286:     void _refreshSortIndex() {
        287:         _refreshList();
        288:         _sortBooks(index);
        289:     }
        290:
        291:     void _refreshSortOrder() {
        292:         _refreshList();
        293:         _sortBooks(order);
        294:     }
        295:
        296:     void _refreshSortColumn() {
        297:         _refreshList();
        298:         _sortBooks(column);
        299:     }
        300:
        301:     void _refreshSortDirection() {
        302:         _refreshList();
        303:         _sortBooks(direction);
        304:     }
        305:
        306:     void _refreshSortTypeIndex() {
        307:         _refreshList();
        308:         _sortBooks(index);
        309:         _sortBooks(type);
        310:     }
        311:
        312:     void _refreshSortTypeOrder() {
        313:         _refreshList();
        314:         _sortBooks(order);
        315:         _sortBooks(type);
        316:     }
        317:
        318:     void _refreshSortTypeColumn() {
        319:         _refreshList();
        320:         _sortBooks(column);
        321:         _sortBooks(type);
        322:     }
        323:
        324:     void _refreshSortTypeDirection() {
        325:         _refreshList();
        326:         _sortBooks(direction);
        327:         _sortBooks(type);
        328:     }
        329:
        330:     void _refreshSortTypeIndexOrder() {
        331:         _refreshList();
        332:         _sortBooks(index);
        333:         _sortBooks(type);
        334:         _sortBooks(order);
        335:     }
        336:
        337:     void _refreshSortTypeIndexColumn() {
        338:         _refreshList();
        339:         _sortBooks(column);
        340:         _sortBooks(type);
        341:         _sortBooks(order);
        342:     }
        343:
        344:     void _refreshSortTypeIndexDirection() {
        345:         _refreshList();
        346:         _sortBooks(direction);
        347:         _sortBooks(type);
        348:         _sortBooks(order);
        349:     }
        350:
        351:     void _refreshSortTypeIndexOrderColumn() {
        352:         _refreshList();
        353:         _sortBooks(column);
        354:         _sortBooks(type);
        355:         _sortBooks(order);
        356:     }
        357:
        358:     void _refreshSortTypeIndexDirectionColumn() {
        359:         _refreshList();
        360:         _sortBooks(direction);
        361:         _sortBooks(type);
        362:         _sortBooks(column);
        363:     }
        364:
        365:     void _refreshSortTypeIndexOrderDirection() {
        366:         _refreshList();
        367:         _sortBooks(direction);
        368:         _sortBooks(column);
        369:         _sortBooks(order);
        370:     }
        371:
        372:     void _refreshSortTypeIndexDirectionOrder() {
        373:         _refreshList();
        374:         _sortBooks(order);
        375:         _sortBooks(column);
        376:         _sortBooks(direction);
        377:     }
        378:
        379:     void _refreshSortTypeIndexOrderDirectionColumn() {
        380:         _refreshList();
        381:         _sortBooks(column);
        382:         _sortBooks(order);
        383:         _sortBooks(direction);
        384:     }
        385:
        386:     void _refreshSortTypeIndexDirectionOrderDirection() {
        387:         _refreshList();
        388:         _sortBooks(direction);
        389:         _sortBooks(order);
        390:         _sortBooks(column);
        391:     }
        392:
        393:     void _refreshSortTypeIndexDirectionOrderDirectionColumn() {
        394:         _refreshList();
        395:         _sortBooks(column);
        396:         _sortBooks(direction);
        397:         _sortBooks(order);
        398:     }
        399:
        400:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirection() {
        401:         _refreshList();
        402:         _sortBooks(direction);
        403:         _sortBooks(column);
        404:         _sortBooks(order);
        405:         _sortBooks(direction);
        406:     }
        407:
        408:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumn() {
        409:         _refreshList();
        410:         _sortBooks(column);
        411:         _sortBooks(direction);
        412:         _sortBooks(order);
        413:         _sortBooks(column);
        414:     }
        415:
        416:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirection() {
        417:         _refreshList();
        418:         _sortBooks(direction);
        419:         _sortBooks(column);
        420:         _sortBooks(order);
        421:         _sortBooks(column);
        422:         _sortBooks(direction);
        423:     }
        424:
        425:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumn() {
        426:         _refreshList();
        427:         _sortBooks(column);
        428:         _sortBooks(direction);
        429:         _sortBooks(order);
        430:         _sortBooks(column);
        431:         _sortBooks(direction);
        432:     }
        433:
        434:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirection() {
        435:         _refreshList();
        436:         _sortBooks(direction);
        437:         _sortBooks(column);
        438:         _sortBooks(order);
        439:         _sortBooks(column);
        440:         _sortBooks(direction);
        441:         _sortBooks(column);
        442:     }
        443:
        444:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        445:         _refreshList();
        446:         _sortBooks(column);
        447:         _sortBooks(direction);
        448:         _sortBooks(order);
        449:         _sortBooks(column);
        450:         _sortBooks(direction);
        451:         _sortBooks(column);
        452:     }
        453:
        454:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        455:         _refreshList();
        456:         _sortBooks(direction);
        457:         _sortBooks(column);
        458:         _sortBooks(order);
        459:         _sortBooks(column);
        460:         _sortBooks(direction);
        461:         _sortBooks(column);
        462:     }
        463:
        464:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        465:         _refreshList();
        466:         _sortBooks(column);
        467:         _sortBooks(direction);
        468:         _sortBooks(order);
        469:         _sortBooks(column);
        470:         _sortBooks(direction);
        471:         _sortBooks(column);
        472:     }
        473:
        474:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        475:         _refreshList();
        476:         _sortBooks(direction);
        477:         _sortBooks(column);
        478:         _sortBooks(order);
        479:         _sortBooks(column);
        480:         _sortBooks(direction);
        481:         _sortBooks(column);
        482:     }
        483:
        484:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        485:         _refreshList();
        486:         _sortBooks(column);
        487:         _sortBooks(direction);
        488:         _sortBooks(order);
        489:         _sortBooks(column);
        490:         _sortBooks(direction);
        491:         _sortBooks(column);
        492:     }
        493:
        494:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        495:         _refreshList();
        496:         _sortBooks(direction);
        497:         _sortBooks(column);
        498:         _sortBooks(order);
        499:         _sortBooks(column);
        500:         _sortBooks(direction);
        490:
        501:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        502:         _refreshList();
        503:         _sortBooks(column);
        504:         _sortBooks(direction);
        505:         _sortBooks(order);
        506:         _sortBooks(column);
        507:         _sortBooks(direction);
        508:         _sortBooks(column);
        509:     }
        510:
        511:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        512:         _refreshList();
        513:         _sortBooks(direction);
        514:         _sortBooks(column);
        515:         _sortBooks(order);
        516:         _sortBooks(column);
        517:         _sortBooks(direction);
        518:         _sortBooks(column);
        519:     }
        520:
        521:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        522:         _refreshList();
        523:         _sortBooks(column);
        524:         _sortBooks(direction);
        525:         _sortBooks(order);
        526:         _sortBooks(column);
        527:         _sortBooks(direction);
        528:         _sortBooks(column);
        529:     }
        530:
        531:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        532:         _refreshList();
        533:         _sortBooks(direction);
        534:         _sortBooks(column);
        535:         _sortBooks(order);
        536:         _sortBooks(column);
        537:         _sortBooks(direction);
        538:         _sortBooks(column);
        539:     }
        540:
        541:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        542:         _refreshList();
        543:         _sortBooks(column);
        544:         _sortBooks(direction);
        545:         _sortBooks(order);
        546:         _sortBooks(column);
        547:         _sortBooks(direction);
        548:         _sortBooks(column);
        549:     }
        550:
        551:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        552:         _refreshList();
        553:         _sortBooks(direction);
        554:         _sortBooks(column);
        555:         _sortBooks(order);
        556:         _sortBooks(column);
        557:         _sortBooks(direction);
        558:         _sortBooks(column);
        559:     }
        560:
        561:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        562:         _refreshList();
        563:         _sortBooks(column);
        564:         _sortBooks(direction);
        565:         _sortBooks(order);
        566:         _sortBooks(column);
        567:         _sortBooks(direction);
        568:         _sortBooks(column);
        569:     }
        570:
        571:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirection() {
        572:         _refreshList();
        573:         _sortBooks(direction);
        574:         _sortBooks(column);
        575:         _sortBooks(order);
        576:         _sortBooks(column);
        577:         _sortBooks(direction);
        578:         _sortBooks(column);
        579:     }
        580:
        581:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        582:         _refreshList();
        583:         _sortBooks(column);
        584:         _sortBooks(direction);
        585:         _sortBooks(order);
        586:         _sortBooks(column);
        587:         _sortBooks(direction);
        588:         _sortBooks(column);
        589:     }
        590:
        591:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        592:         _refreshList();
        593:         _sortBooks(direction);
        594:         _sortBooks(column);
        595:         _sortBooks(order);
        596:         _sortBooks(column);
        597:         _sortBooks(direction);
        598:         _sortBooks(column);
        599:     }
        600:
        601:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        602:         _refreshList();
        603:         _sortBooks(column);
        604:         _sortBooks(direction);
        605:         _sortBooks(order);
        606:         _sortBooks(column);
        607:         _sortBooks(direction);
        608:         _sortBooks(column);
        609:     }
        610:
        611:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        612:         _refreshList();
        613:         _sortBooks(direction);
        614:         _sortBooks(column);
        615:         _sortBooks(order);
        616:         _sortBooks(column);
        617:         _sortBooks(direction);
        618:         _sortBooks(column);
        619:     }
        620:
        621:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        622:         _refreshList();
        623:         _sortBooks(column);
        624:         _sortBooks(direction);
        625:         _sortBooks(order);
        626:         _sortBooks(column);
        627:         _sortBooks(direction);
        628:         _sortBooks(column);
        629:     }
        630:
        631:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        632:         _refreshList();
        633:         _sortBooks(direction);
        634:         _sortBooks(column);
        635:         _sortBooks(order);
        636:         _sortBooks(column);
        637:         _sortBooks(direction);
        638:         _sortBooks(column);
        639:     }
        640:
        641:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        642:         _refreshList();
        643:         _sortBooks(column);
        644:         _sortBooks(direction);
        645:         _sortBooks(order);
        646:         _sortBooks(column);
        647:         _sortBooks(direction);
        648:         _sortBooks(column);
        649:     }
        650:
        651:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        652:         _refreshList();
        653:         _sortBooks(direction);
        654:         _sortBooks(column);
        655:         _sortBooks(order);
        656:         _sortBooks(column);
        657:         _sortBooks(direction);
        658:         _sortBooks(column);
        659:     }
        660:
        661:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        662:         _refreshList();
        663:         _sortBooks(column);
        664:         _sortBooks(direction);
        665:         _sortBooks(order);
        666:         _sortBooks(column);
        667:         _sortBooks(direction);
        668:         _sortBooks(column);
        669:     }
        670:
        671:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        672:         _refreshList();
        673:         _sortBooks(direction);
        674:         _sortBooks(column);
        675:         _sortBooks(order);
        676:         _sortBooks(column);
        677:         _sortBooks(direction);
        678:         _sortBooks(column);
        679:     }
        680:
        681:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        682:         _refreshList();
        683:         _sortBooks(column);
        684:         _sortBooks(direction);
        685:         _sortBooks(order);
        686:         _sortBooks(column);
        687:         _sortBooks(direction);
        688:         _sortBooks(column);
        689:     }
        690:
        691:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        692:         _refreshList();
        693:         _sortBooks(direction);
        694:         _sortBooks(column);
        695:         _sortBooks(order);
        696:         _sortBooks(column);
        697:         _sortBooks(direction);
        698:         _sortBooks(column);
        699:     }
        700:
        701:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        702:         _refreshList();
        703:         _sortBooks(column);
        704:         _sortBooks(direction);
        705:         _sortBooks(order);
        706:         _sortBooks(column);
        707:         _sortBooks(direction);
        708:         _sortBooks(column);
        709:     }
        710:
        711:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        712:         _refreshList();
        713:         _sortBooks(direction);
        714:         _sortBooks(column);
        715:         _sortBooks(order);
        716:         _sortBooks(column);
        717:         _sortBooks(direction);
        718:         _sortBooks(column);
        719:     }
        720:
        721:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        722:         _refreshList();
        723:         _sortBooks(column);
        724:         _sortBooks(direction);
        725:         _sortBooks(order);
        726:         _sortBooks(column);
        727:         _sortBooks(direction);
        728:         _sortBooks(column);
        729:     }
        730:
        731:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        732:         _refreshList();
        733:         _sortBooks(direction);
        734:         _sortBooks(column);
        735:         _sortBooks(order);
        736:         _sortBooks(column);
        737:         _sortBooks(direction);
        738:         _sortBooks(column);
        739:     }
        740:
        741:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        742:         _refreshList();
        743:         _sortBooks(column);
        744:         _sortBooks(direction);
        745:         _sortBooks(order);
        746:         _sortBooks(column);
        747:         _sortBooks(direction);
        748:         _sortBooks(column);
        749:     }
        750:
        751:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        752:         _refreshList();
        753:         _sortBooks(direction);
        754:         _sortBooks(column);
        755:         _sortBooks(order);
        756:         _sortBooks(column);
        757:         _sortBooks(direction);
        758:         _sortBooks(column);
        759:     }
        760:
        761:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        762:         _refreshList();
        763:         _sortBooks(column);
        764:         _sortBooks(direction);
        765:         _sortBooks(order);
        766:         _sortBooks(column);
        767:         _sortBooks(direction);
        768:         _sortBooks(column);
        769:     }
        770:
        771:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        772:         _refreshList();
        773:         _sortBooks(direction);
        774:         _sortBooks(column);
        775:         _sortBooks(order);
        776:         _sortBooks(column);
        777:         _sortBooks(direction);
        778:         _sortBooks(column);
        779:     }
        780:
        781:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        782:         _refreshList();
        783:         _sortBooks(column);
        784:         _sortBooks(direction);
        785:         _sortBooks(order);
        786:         _sortBooks(column);
        787:         _sortBooks(direction);
        788:         _sortBooks(column);
        789:     }
        790:
        791:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        792:         _refreshList();
        793:         _sortBooks(direction);
        794:         _sortBooks(column);
        795:         _sortBooks(order);
        796:         _sortBooks(column);
        797:         _sortBooks(direction);
        798:         _sortBooks(column);
        799:     }
        800:
        801:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        802:         _refreshList();
        803:         _sortBooks(column);
        804:         _sortBooks(direction);
        805:         _sortBooks(order);
        806:         _sortBooks(column);
        807:         _sortBooks(direction);
        808:         _sortBooks(column);
        809:     }
        810:
        811:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        812:         _refreshList();
        813:         _sortBooks(direction);
        814:         _sortBooks(column);
        815:         _sortBooks(order);
        816:         _sortBooks(column);
        817:         _sortBooks(direction);
        818:         _sortBooks(column);
        819:     }
        820:
        821:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        822:         _refreshList();
        823:         _sortBooks(column);
        824:         _sortBooks(direction);
        825:         _sortBooks(order);
        826:         _sortBooks(column);
        827:         _sortBooks(direction);
        828:         _sortBooks(column);
        829:     }
        830:
        831:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        832:         _refreshList();
        833:         _sortBooks(direction);
        834:         _sortBooks(column);
        835:         _sortBooks(order);
        836:         _sortBooks(column);
        837:         _sortBooks(direction);
        838:         _sortBooks(column);
        839:     }
        840:
        841:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        842:         _refreshList();
        843:         _sortBooks(column);
        844:         _sortBooks(direction);
        845:         _sortBooks(order);
        846:         _sortBooks(column);
        847:         _sortBooks(direction);
        848:         _sortBooks(column);
        849:     }
        850:
        851:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        852:         _refreshList();
        853:         _sortBooks(direction);
        854:         _sortBooks(column);
        855:         _sortBooks(order);
        856:         _sortBooks(column);
        857:         _sortBooks(direction);
        858:         _sortBooks(column);
        859:     }
        860:
        861:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        862:         _refreshList();
        863:         _sortBooks(column);
        864:         _sortBooks(direction);
        865:         _sortBooks(order);
        866:         _sortBooks(column);
        867:         _sortBooks(direction);
        868:         _sortBooks(column);
        869:     }
        870:
        871:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        872:         _refreshList();
        873:         _sortBooks(direction);
        874:         _sortBooks(column);
        875:         _sortBooks(order);
        876:         _sortBooks(column);
        877:         _sortBooks(direction);
        878:         _sortBooks(column);
        879:     }
        880:
        881:     void _refreshSortTypeIndexDirectionOrderDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumnDirectionColumn() {
        882:         _refreshList();
        883:         _sortBooks(column);
        884:         _sortBooks(direction);
        885:         _sortBooks(order);
        886:         _sortBooks(column);
        887:         _sortBooks(direction);
        888:         _sortBooks(column);
        889:     }
        890:
        891:     void
```



7. Output UI Flutter

UI hanya menampilkan jumlah buku dan tombol refresh.



Ini sudah cukup sesuai perintah, karena UI tidak diminta menampilkan detail buku.

KESIMPULAN PRAKTIKUM

Pada praktikum ini dapat disimpulkan bahwa proses integrasi antara **Flutter** dan **Supabase** dapat berjalan dengan baik untuk membangun sistem **perpustakaan digital sederhana**. Supabase berhasil digunakan sebagai backend yang menyediakan layanan database berbasis PostgreSQL dengan akses mudah melalui API.

Melalui praktikum ini, beberapa hal yang berhasil dicapai adalah:

1. **Berhasil membuat database dan tabel perpustakaan di Supabase**
dengan struktur kolom sesuai kebutuhan, seperti judul, penulis, tahun_terbit, genre, dan created_at.
2. **Aplikasi Flutter berhasil terhubung ke Supabase**
melalui proses inisialisasi menggunakan Supabase.initialize() dengan URL dan anon key.
3. **Operasi CRUD (Create and Read) dapat berjalan dengan benar**
 - o Data buku dapat ditambahkan ke tabel Supabase menggunakan perintah insert.
 - o Data buku dapat dibaca kembali menggunakan perintah select dan ditampilkan di terminal.
4. **Data berhasil tampil dalam format berbeda**
baik berupa detail tiap buku, daftar keseluruhan, maupun jumlah total buku yang tersedia.
5. **Flutter mampu menampilkan hasil operasi secara real-time**
yaitu dengan menampilkan jumlah total buku pada UI, dan detail buku ditampilkan pada console-terminal.

Secara keseluruhan, praktikum ini menunjukkan bahwa **Supabase adalah alternatif backend yang powerful, mudah digunakan, dan sangat kompatibel dengan Flutter** untuk membuat aplikasi modern berbasis cloud.



Praktikum ini juga melatih pemahaman mengenai pengolahan data (CRUD), integrasi API, dan penggunaan model data dalam aplikasi Flutter.