Nama : Muhammad Najmy Wardhana

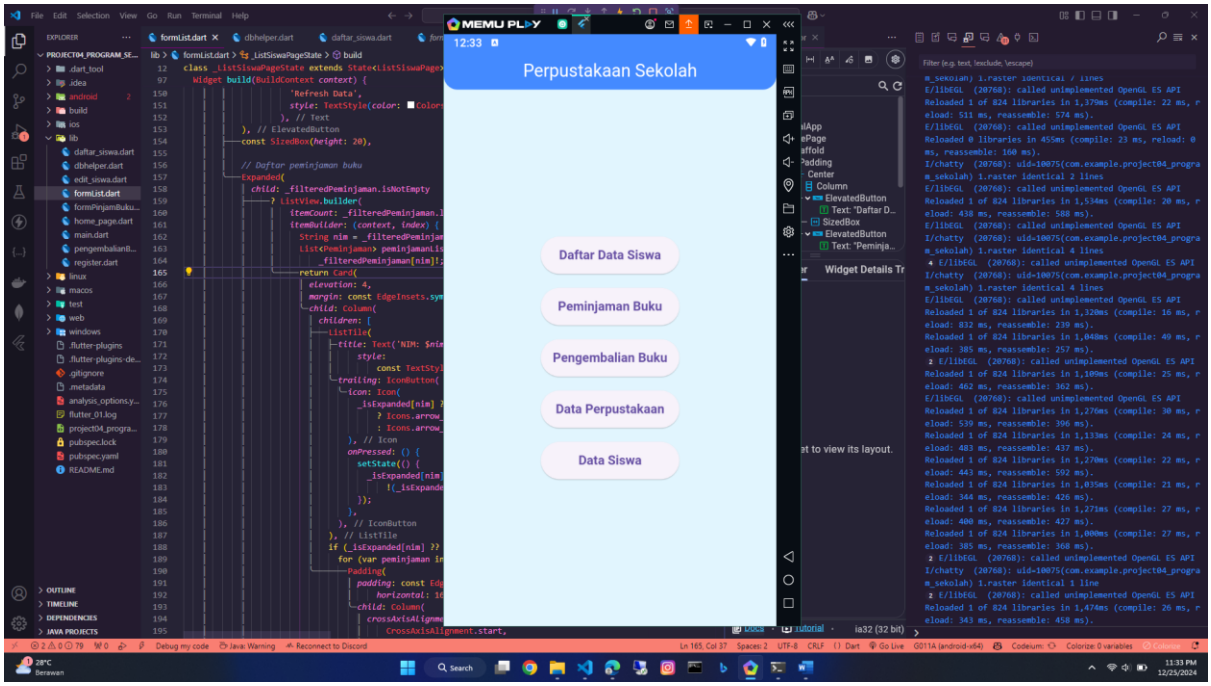
NPM : 2210010212

Kelas : 5B TI Reguler Pagi BJB

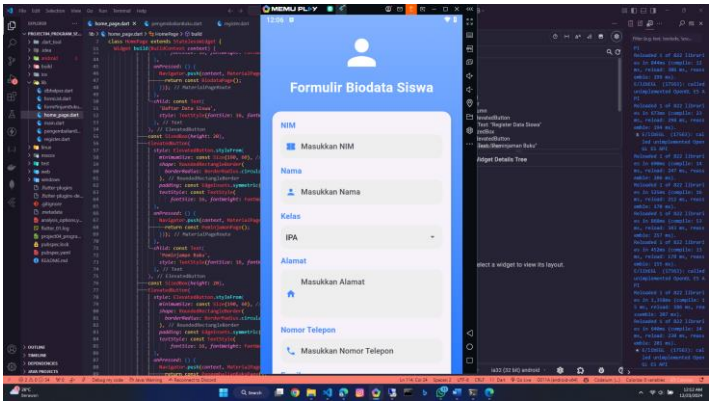
Nomor Akhir Absen 11 (1)

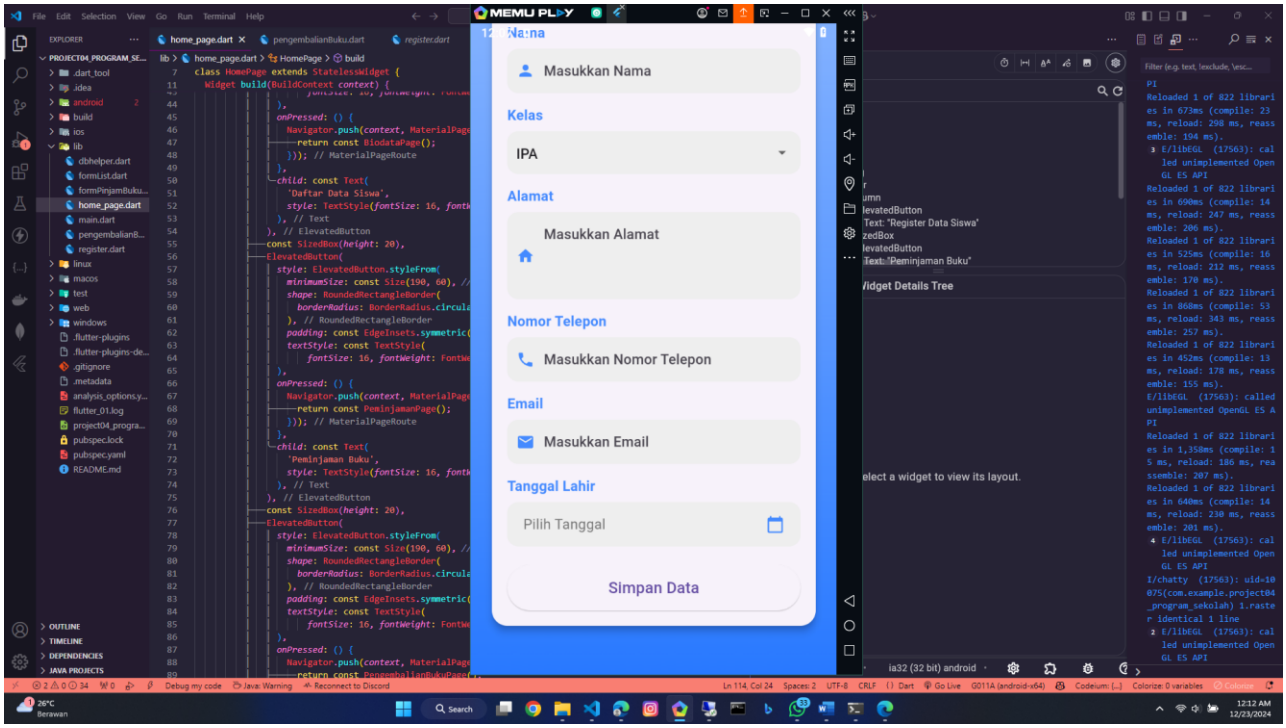
Topik Program Sekolah;
Perpustakaan

Screenshot

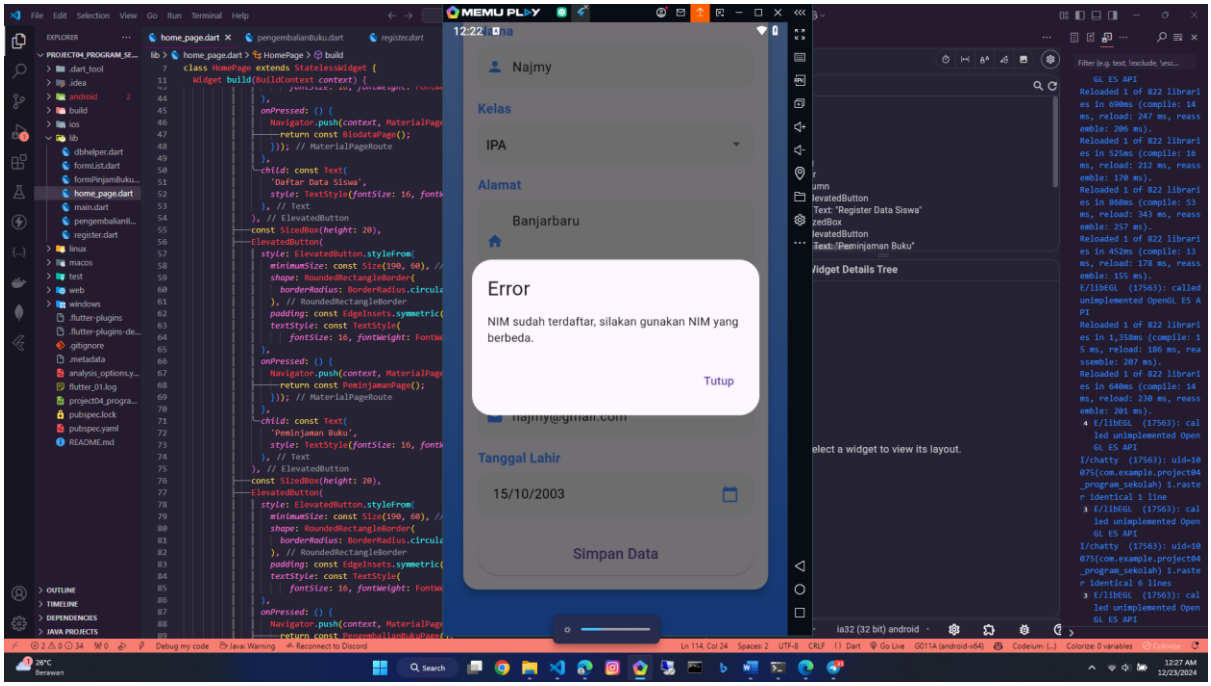


- Saya akan pilih *Register Data Siswa*

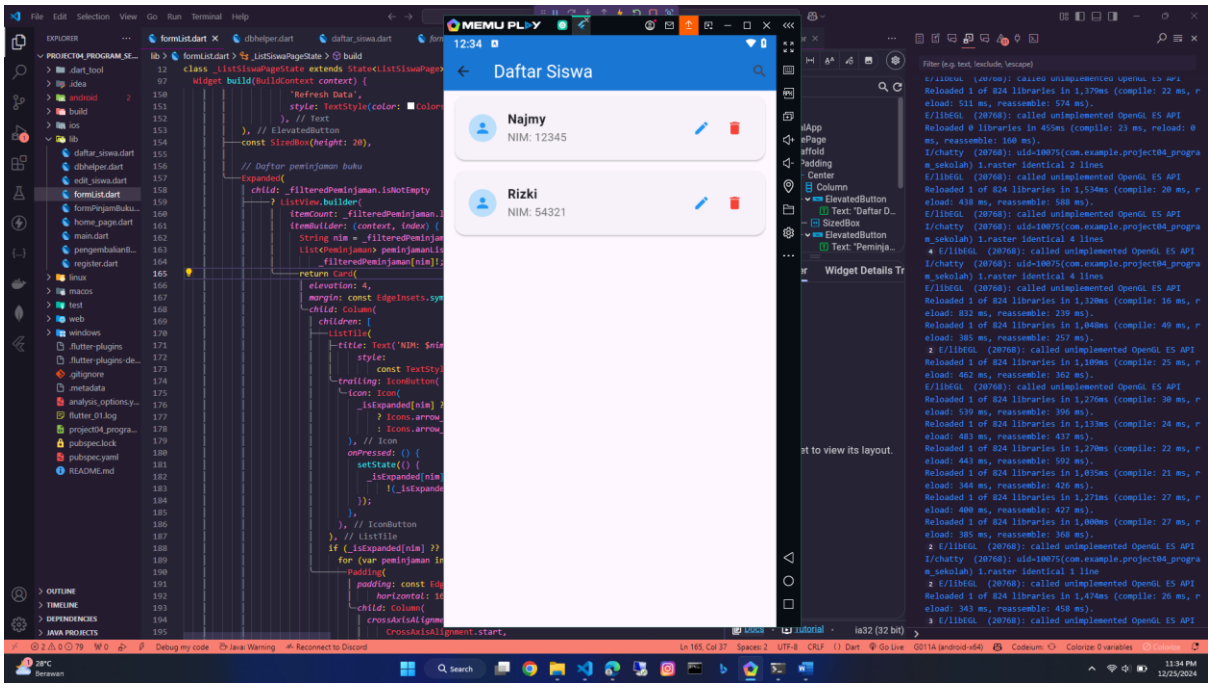




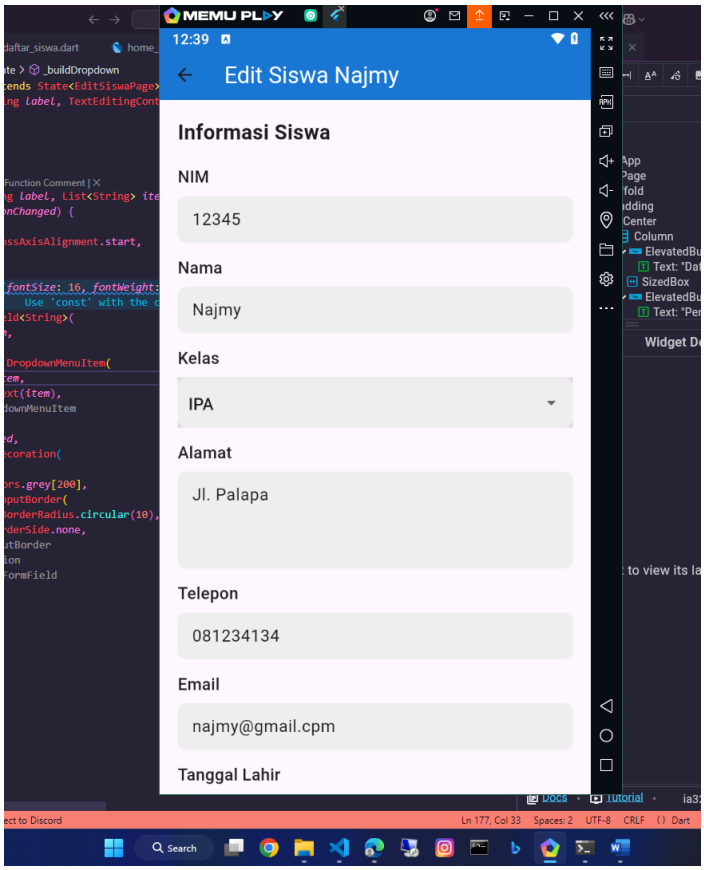
- Saya akan input NIM yang sama sebelumnya sudah didaftarkan

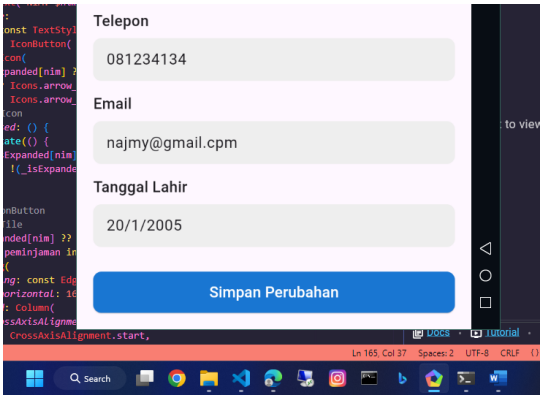


- Kembali ke halaman utama, lalu masuk ke *Data Siswa,* disini kita bisa *Menghapus Data atau Mengedit (update) data*

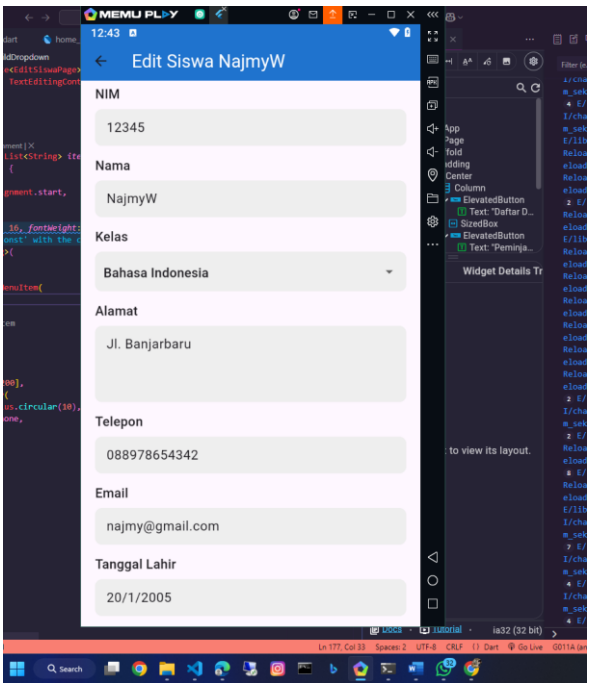


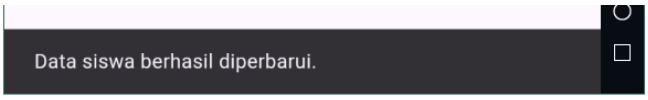
- Saya akan *Edit Data Najmy,* saya akan edit Seluruhnya



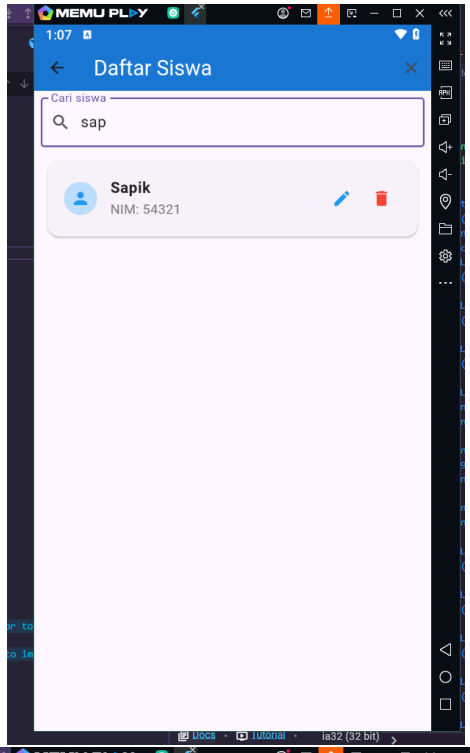


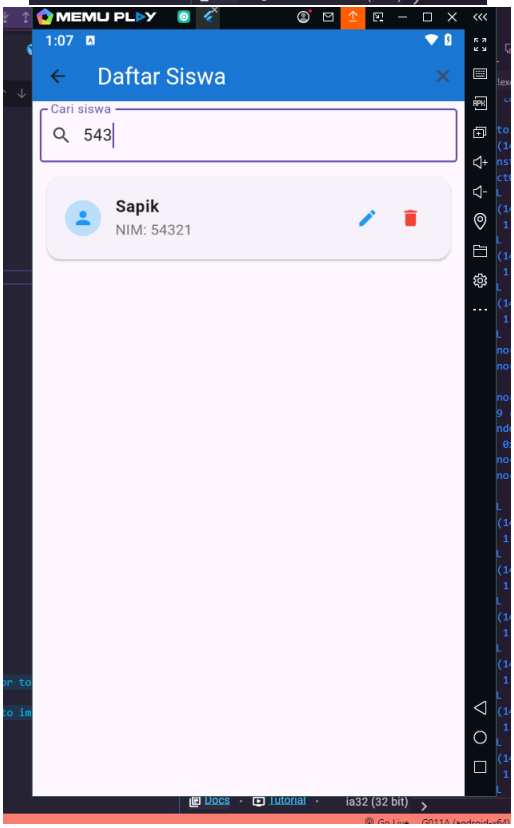
- Data yang sudah update akan tampil begini

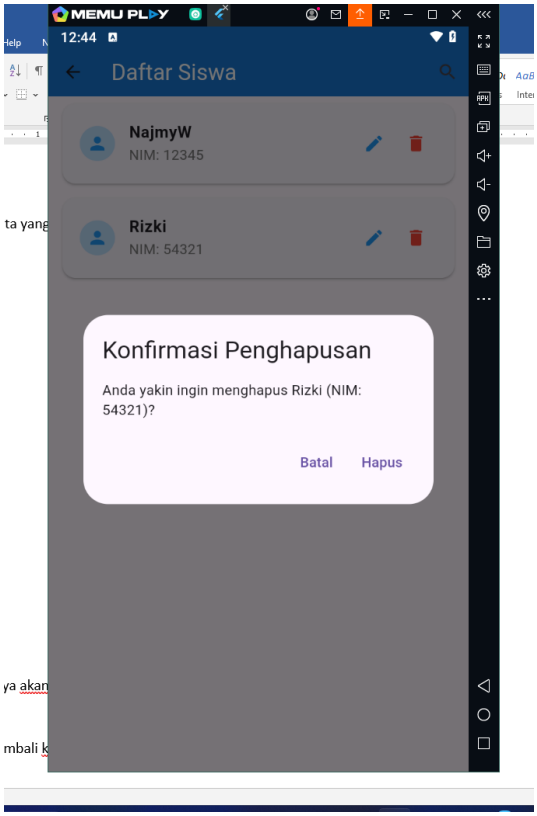




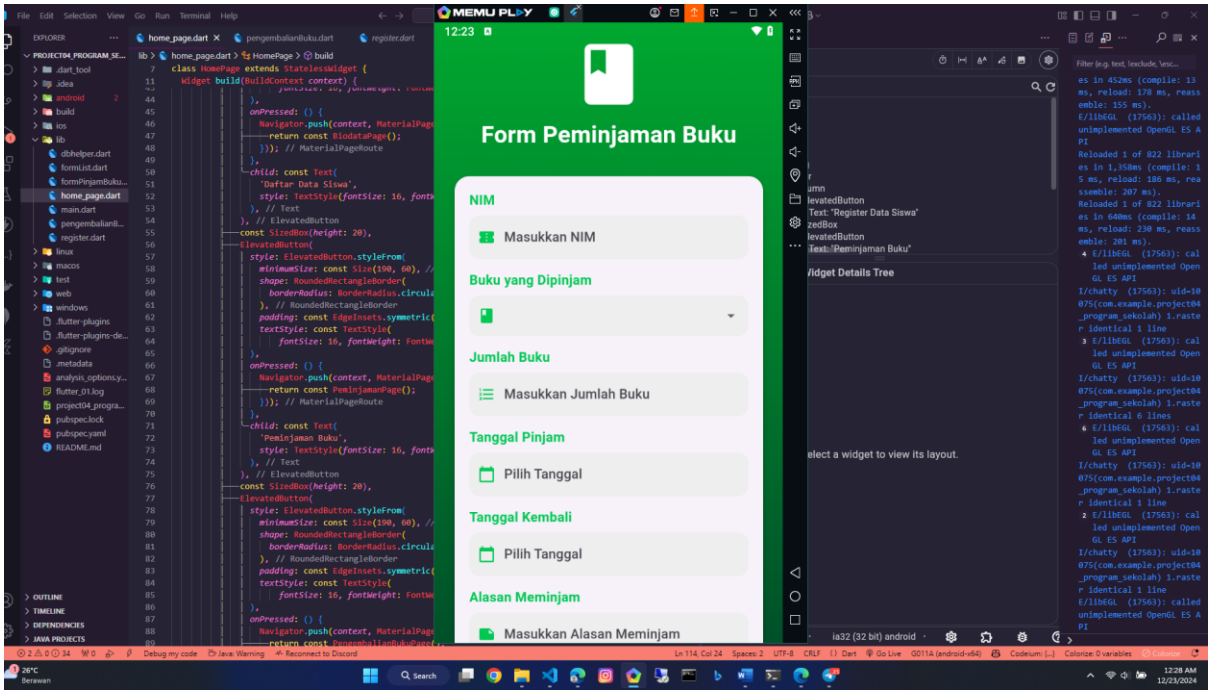
- Saya akan mencari Data sesuai Nama atau NIM, kemudian akan *Hapus* salah satu Datanya.

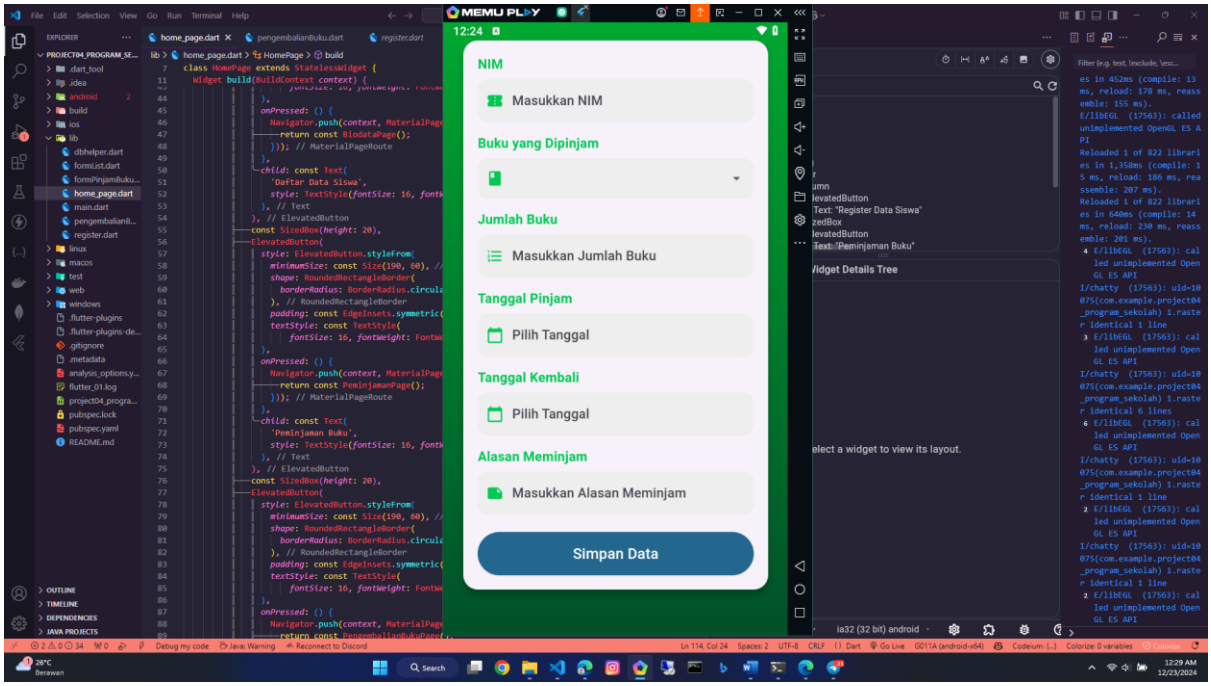




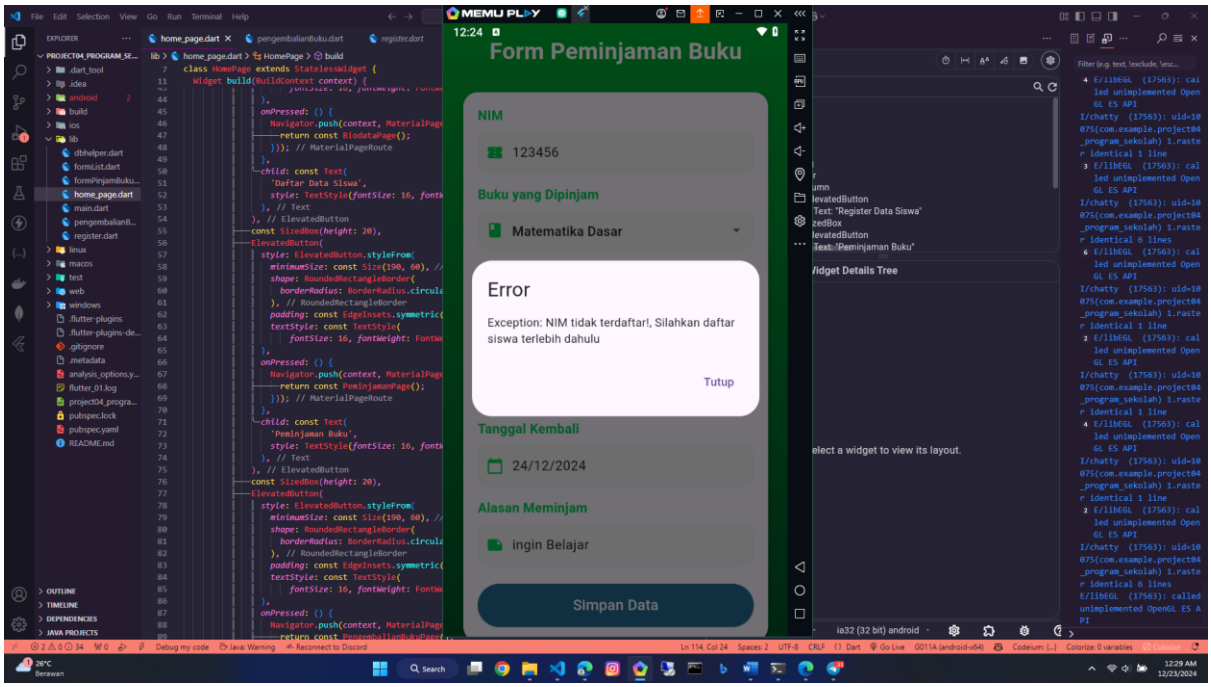


- Kembali ke halaman utama, lalu masuk ke *Peminjaman Buku*

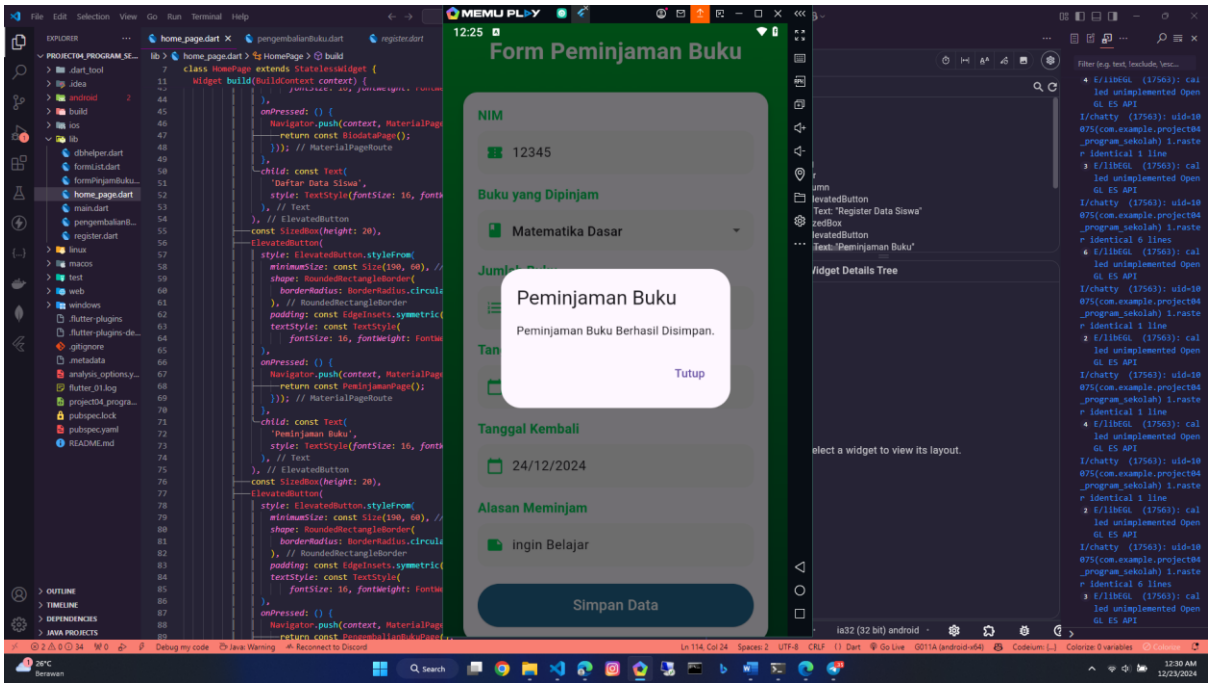




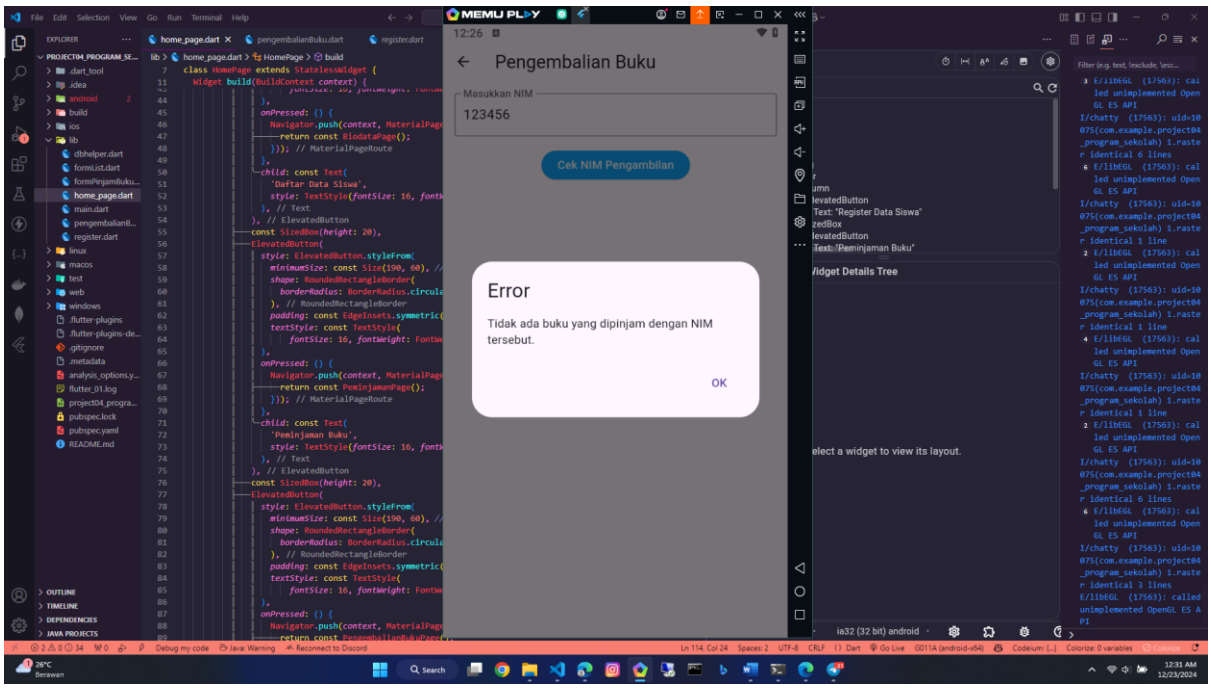
- Saya akan mengisi seluruh inputan dan mengisi NIM yang belum terdaftar



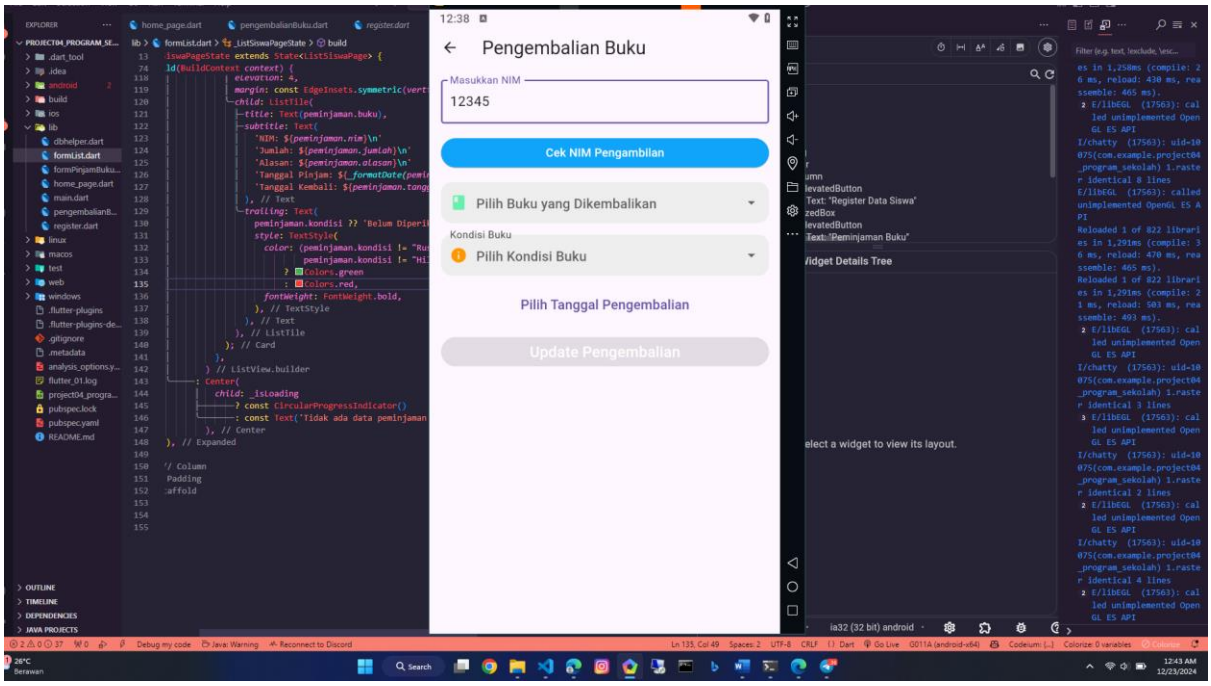
- Saya mengisi kembali dengan NIM yang sudah didaftarkan



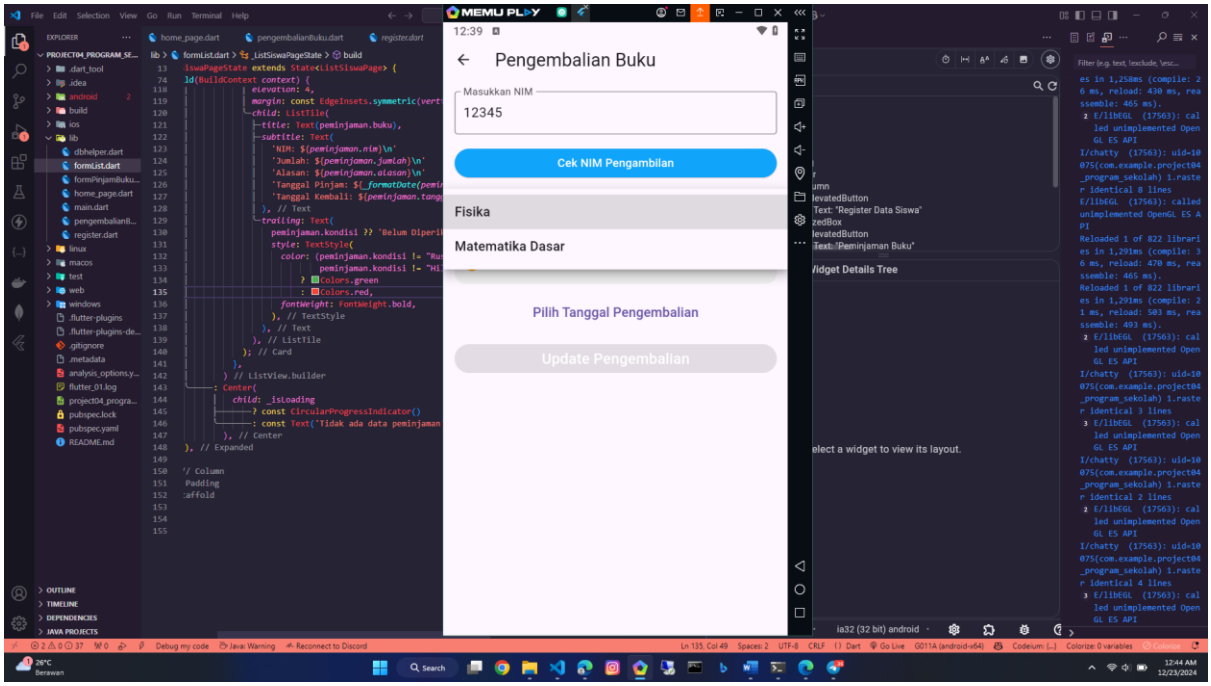
- Kembali ke Halaman Utama, lalu memilih *Form Pengembalian Buku* dan mengisi NIM yang tidak terdaftar/belum ada peminjaman buku



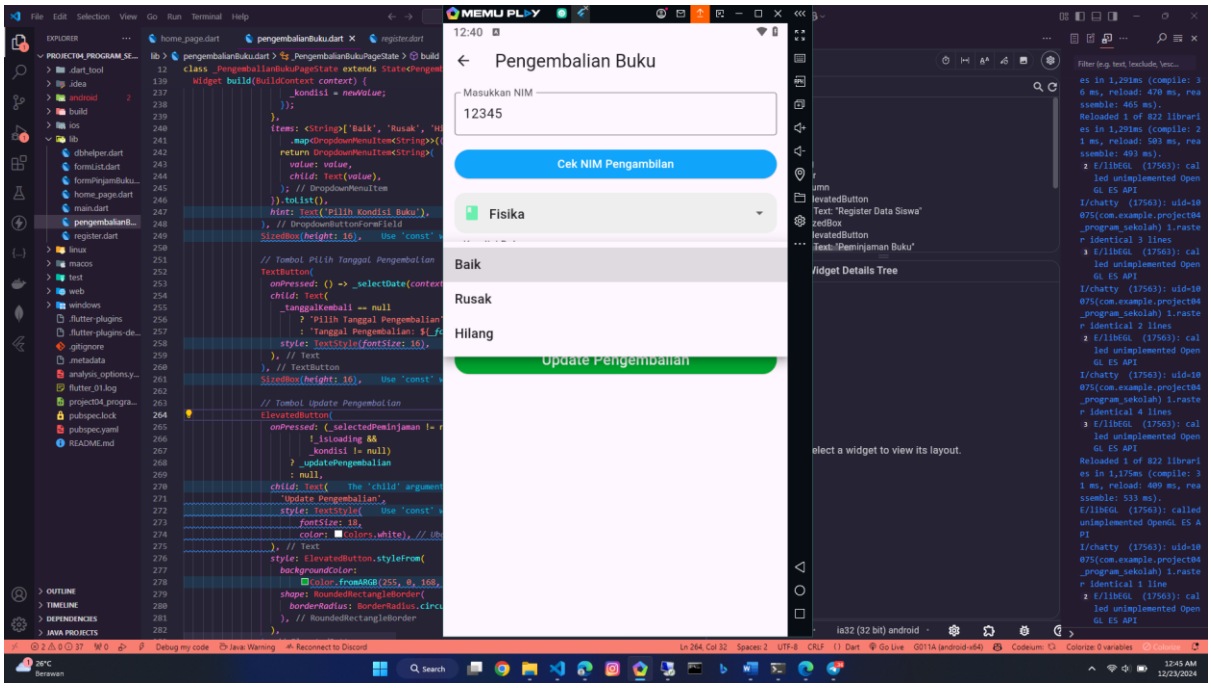
- Saya Kembali memasukan NIM yang sudah ada pinjaman buku

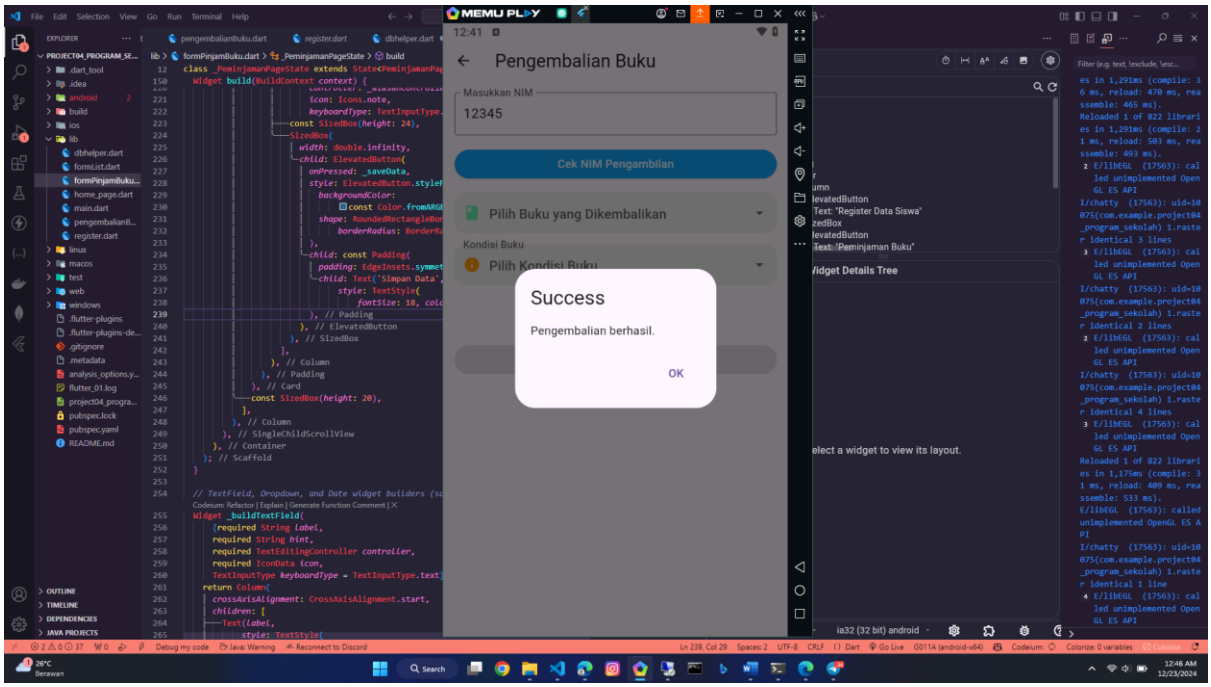


- Terlihat ada 2 Buku yang saya pinjam diperpustakaan sekolah.

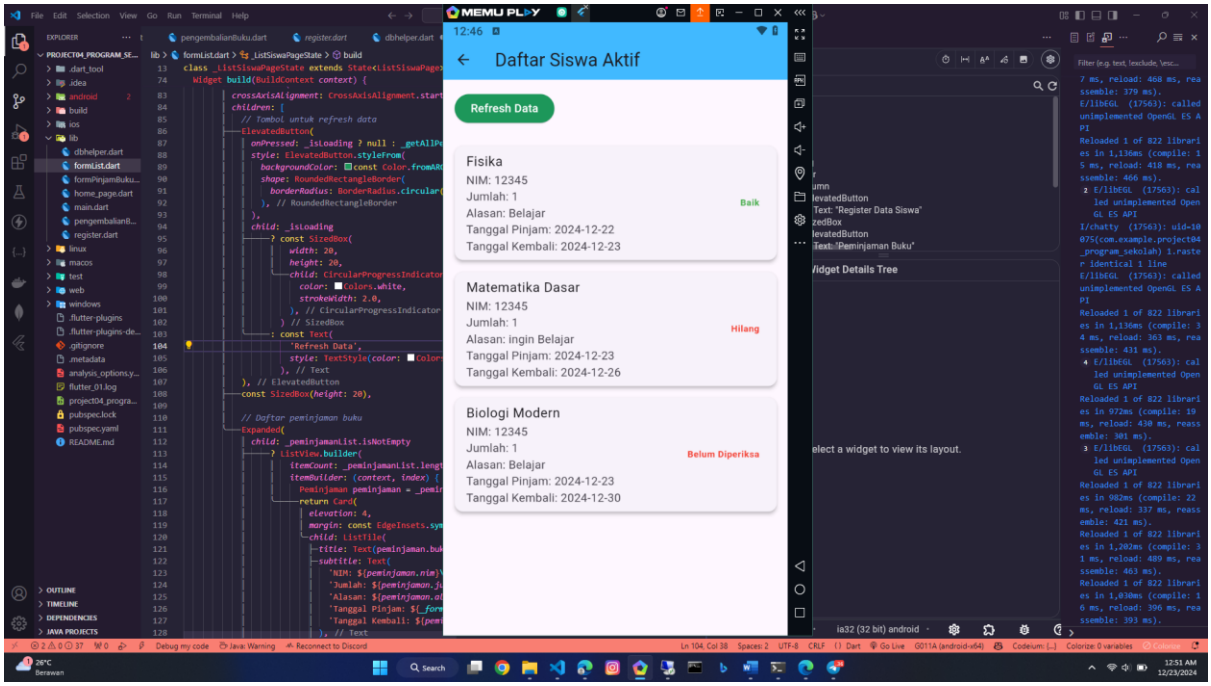


- Saya memilih salah satu buku yang ingin saya kembalikan ke perpustakaan, lalu saya input *Tanggal Pengembalian, Kondisi Buku* dan klik tombol *Update Pengembalian*

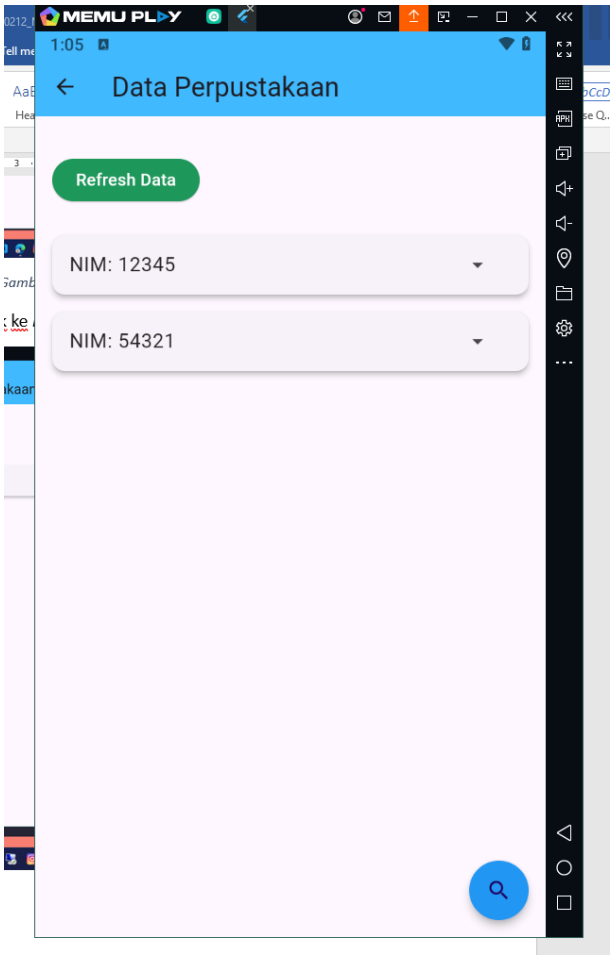




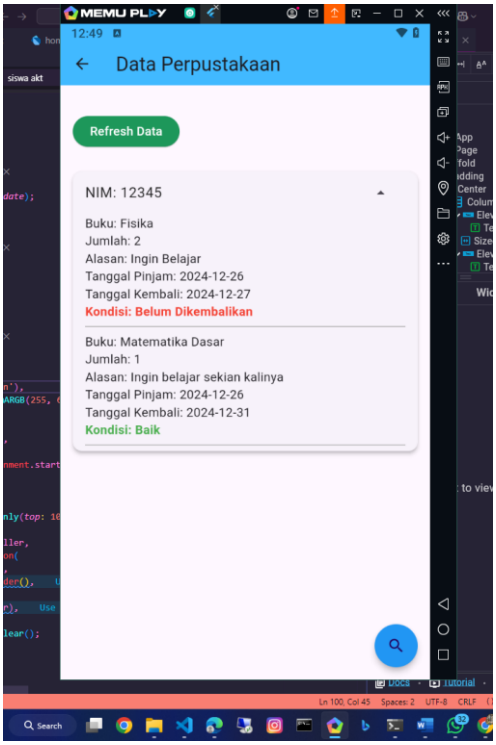
- Kembali ke Halaman Utama, saya akan memilih *Form Daftar Siswa Aktif.*

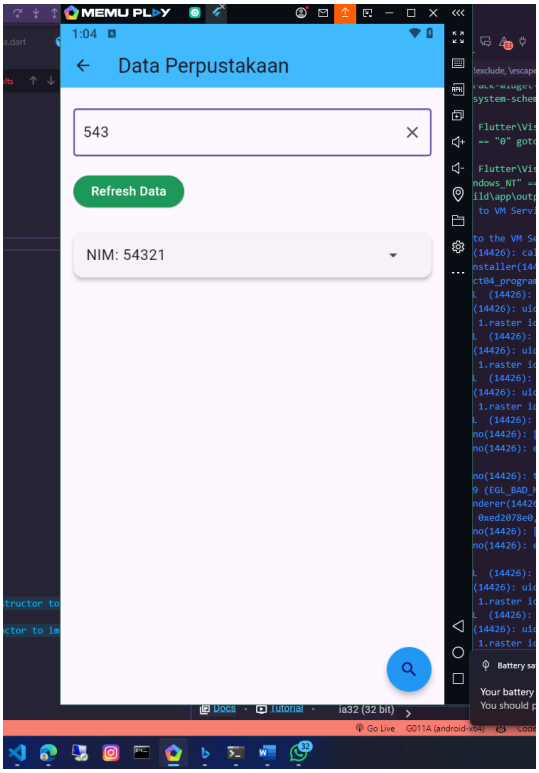


- Kembali ke halaman utama, lalu masuk ke *Data Perpustakaan*

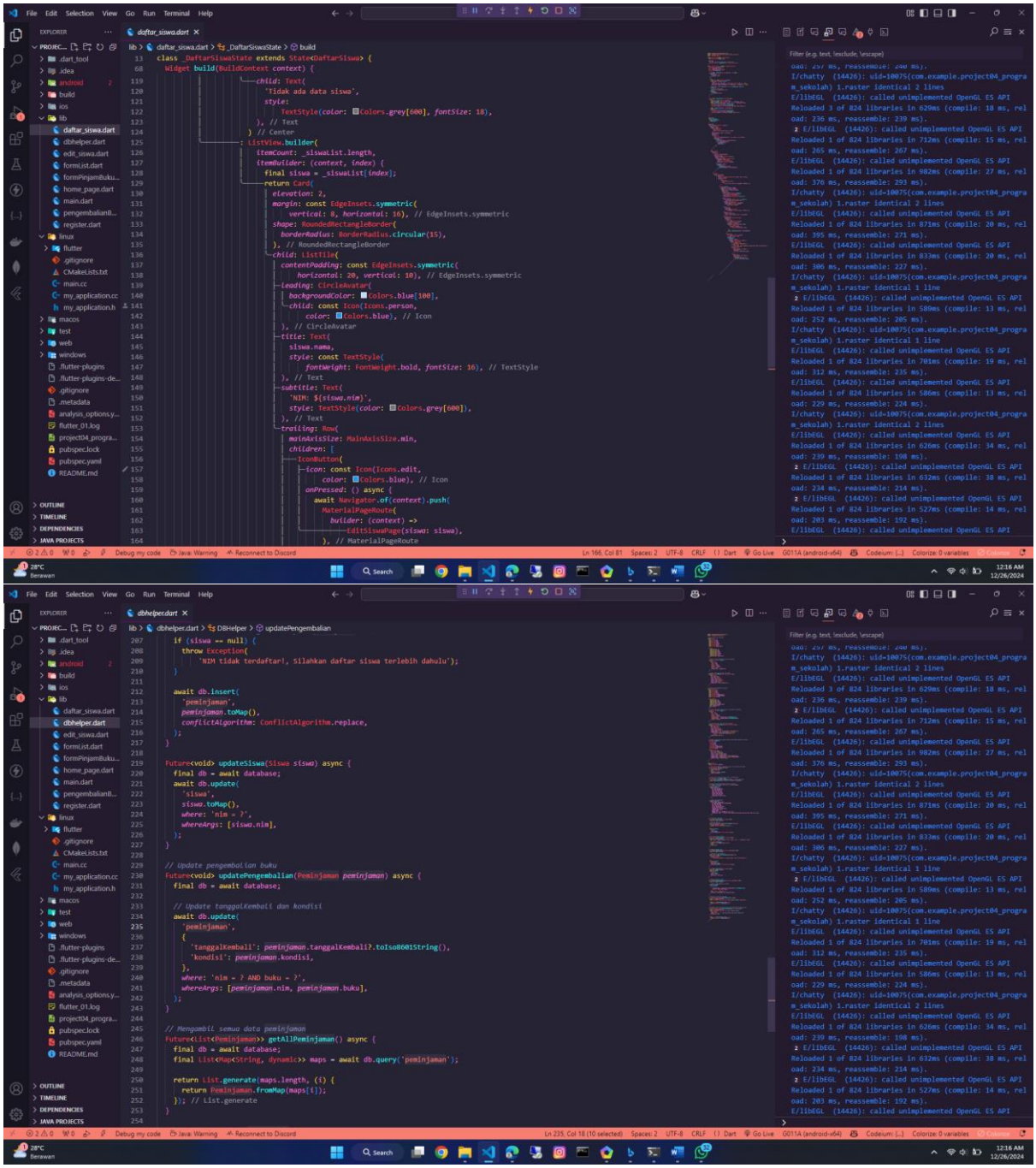


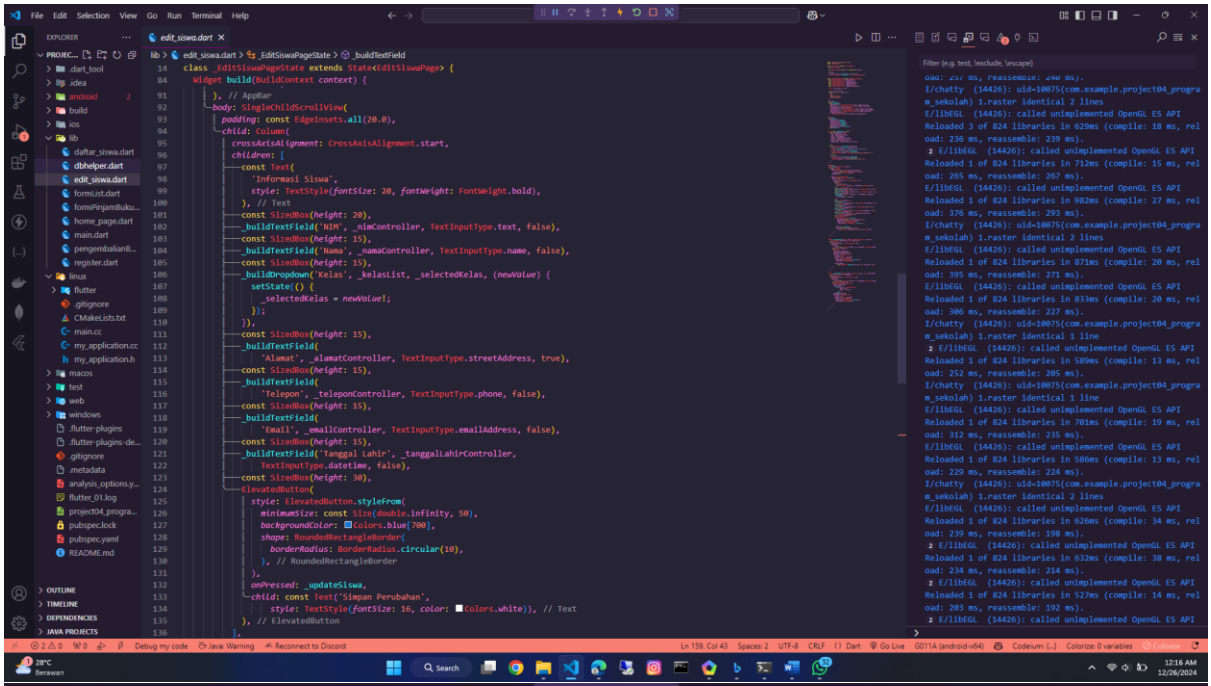
- Jika saya klik user NIM tersebut, maka akan tampil seluruh kegiatan Siswa tersebut. Lalu saya akan mencoba search data sesuai NIM.



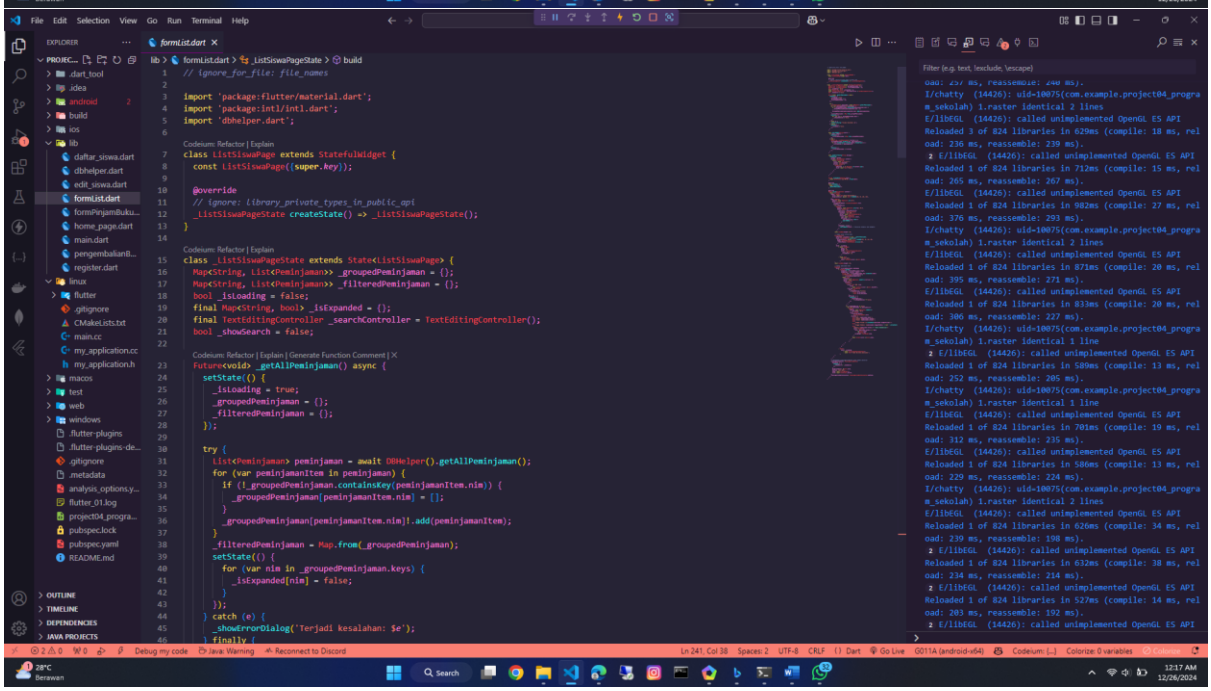


Code

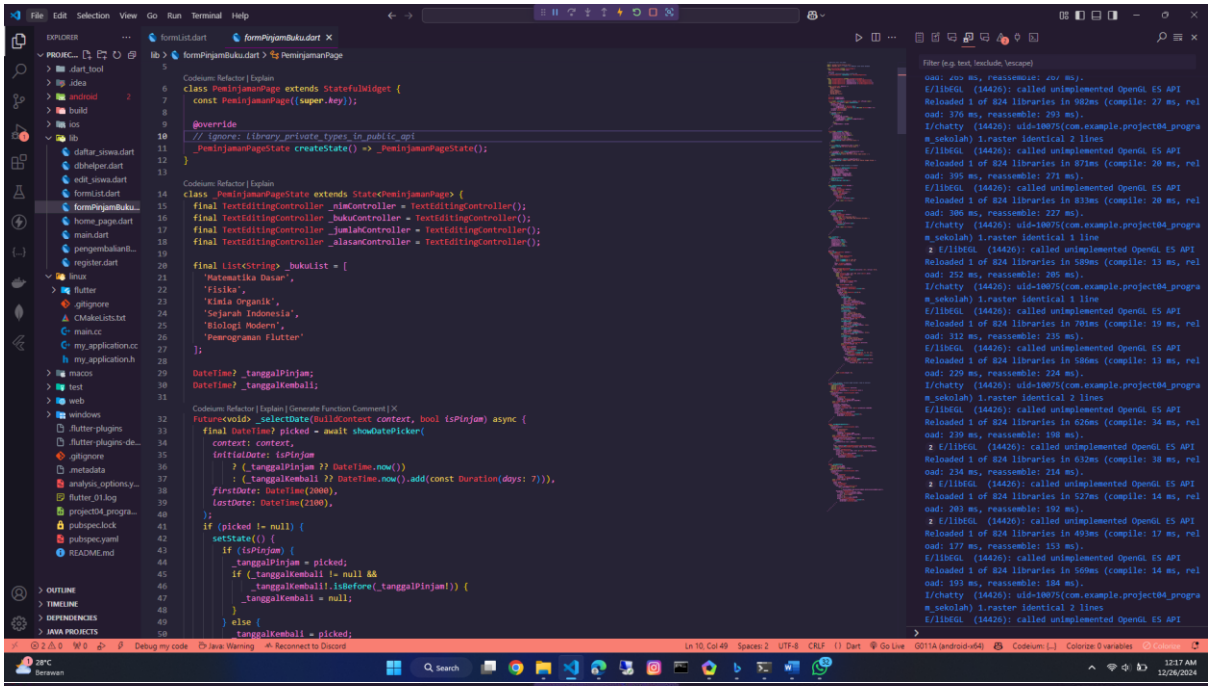




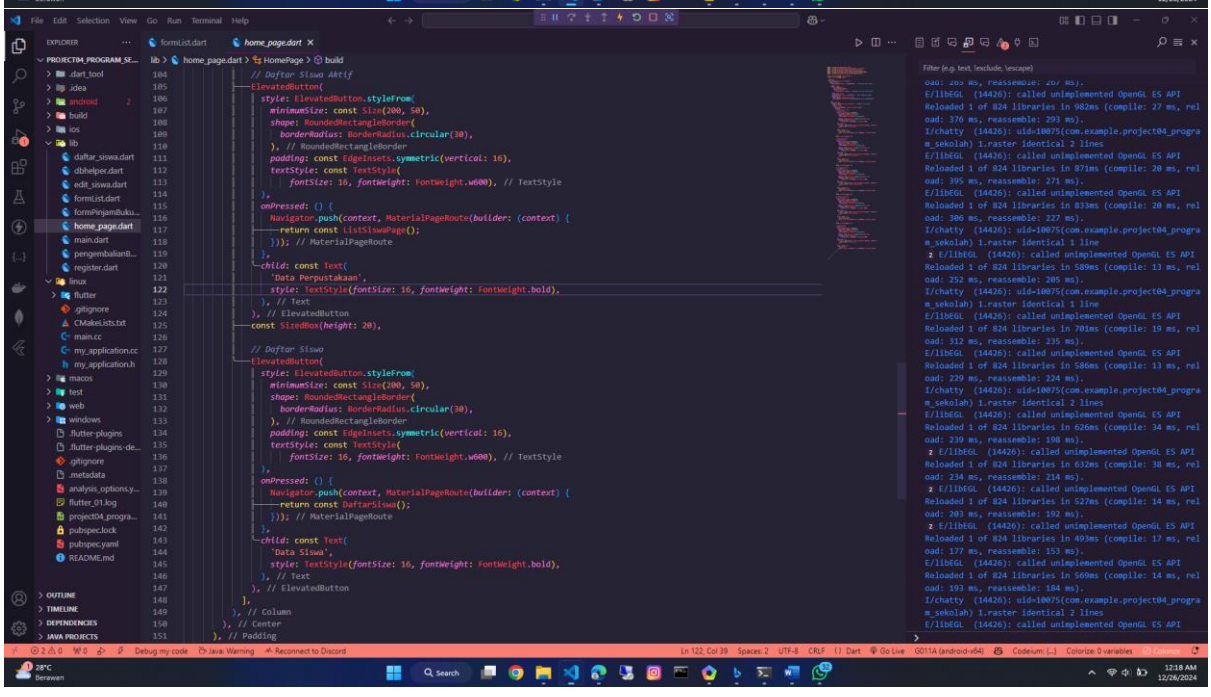
```
class _EditSiswaPageState extends State<EditSiswaPage> {
  Widget build(BuildContext context) {
    ), // AppBar
    body: SingleChildScrollView(
      padding: const EdgeInsets.all(20.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Text(
            'Informasi Siswa',
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ), // Text
          const SizedBox(height: 20),
          _buildTextField('NIM', _nimController, TextInputType.text, false),
          const SizedBox(height: 15),
          _buildTextField('Nama', _namaController, TextInputType.name, false),
          const SizedBox(height: 15),
          _buildDropdown('Kelas', _kelasList, _selectedKelas, (newValue) {
            setState(() {
              _selectedKelas = newValue;
            });
          }),
          const SizedBox(height: 15),
          _buildTextField(
            'Alamat', _alamatController, TextInputType.streetAddress, true),
          const SizedBox(height: 15),
          _buildTextField(
            'Telepon', _teleponController, TextInputType.phone, false),
          const SizedBox(height: 15),
          _buildTextField(
            'Email', _emailController, TextInputType.emailAddress, false),
          const SizedBox(height: 15),
          _buildTextField('Tanggal Lahir', _tanggalLahirController,
              TextInputType.datetime, false),
          const SizedBox(height: 30),
          ElevatedButton(
            style: ElevatedButton.styleFrom(
              minimumSize: const Size(double.infinity, 50),
              backgroundColor: Colors.blue[700],
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ), // RoundedRectangleBorder
            ),
            onPressed: _updateSiswa,
            child: const Text('Simpan Perubahan',
                style: TextStyle(fontSize: 16, color: Colors.white)), // Text
          ), // ElevatedButton
        ],
```



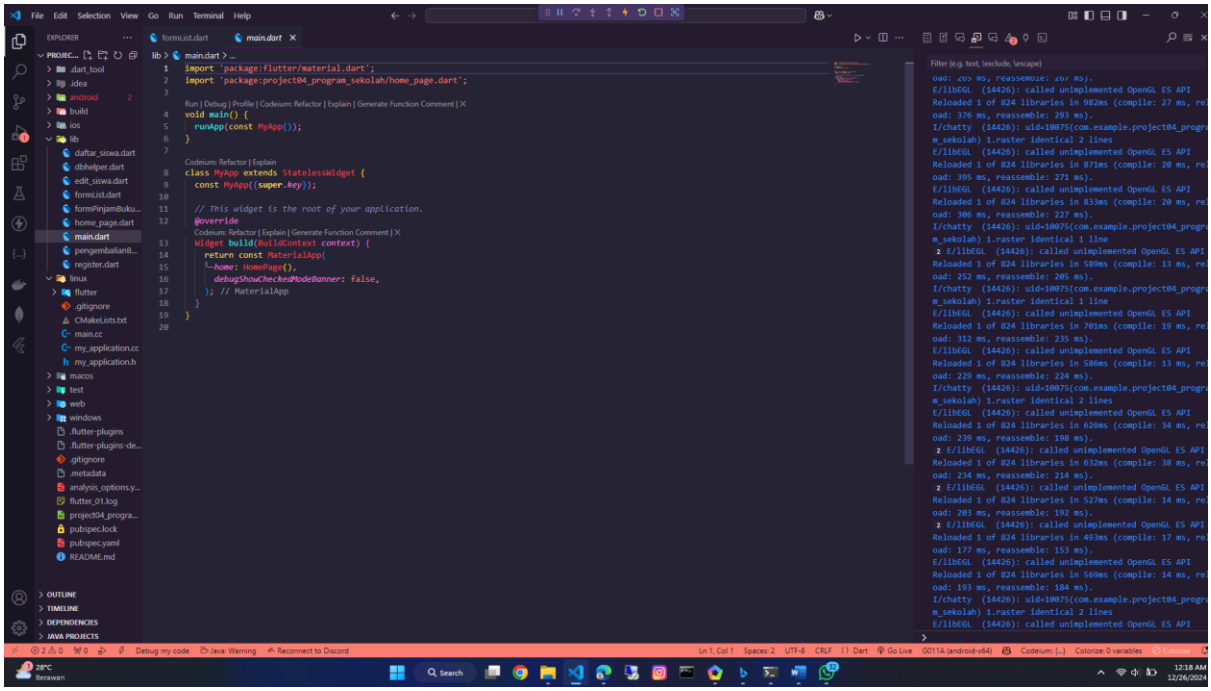
```
// ignore_for_file: file_names

import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'dbhelper.dart';

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

  @override
  // ignore: library_private_types_in_public_api
  _ListSiswaPageState createState() => _ListSiswaPageState();
}

class _ListSiswaPageState extends State<ListSiswaPage> {
  Map<String, List<Peminjaman>> _groupedPeminjaman = {};
  Map<String, List<Peminjaman>> _filteredPeminjaman = {};
  bool _isLoading = false;
  final Map<String, bool> _isExpanded = {};
  final TextEditingController _searchController = TextEditingController();
  bool _showSearch = false;

  Future<void> _getAllPeminjaman() async {
    setState(() {
      _isLoading = true;
      _groupedPeminjaman = {};
      _filteredPeminjaman = {};
    });

    try {
      List<Peminjaman> peminjaman = await DBHelper().getAllPeminjaman();
      for (var peminjamanItem in peminjaman) {
        if (!_groupedPeminjaman.containsKey(peminjamanItem.nim)) {
          _groupedPeminjaman[peminjamanItem.nim] = [];
        }
        _groupedPeminjaman[peminjamanItem.nim]!.add(peminjamanItem);
      }
      _filteredPeminjaman = Map.from(_groupedPeminjaman);
      setState(() {
        for (var nim in _groupedPeminjaman.keys) {
          _isExpanded[nim] = false;
        }
      });
    } catch (e) {
      _showErrorDialog('Terjadi kesalahan: $e');
    } finally {
```



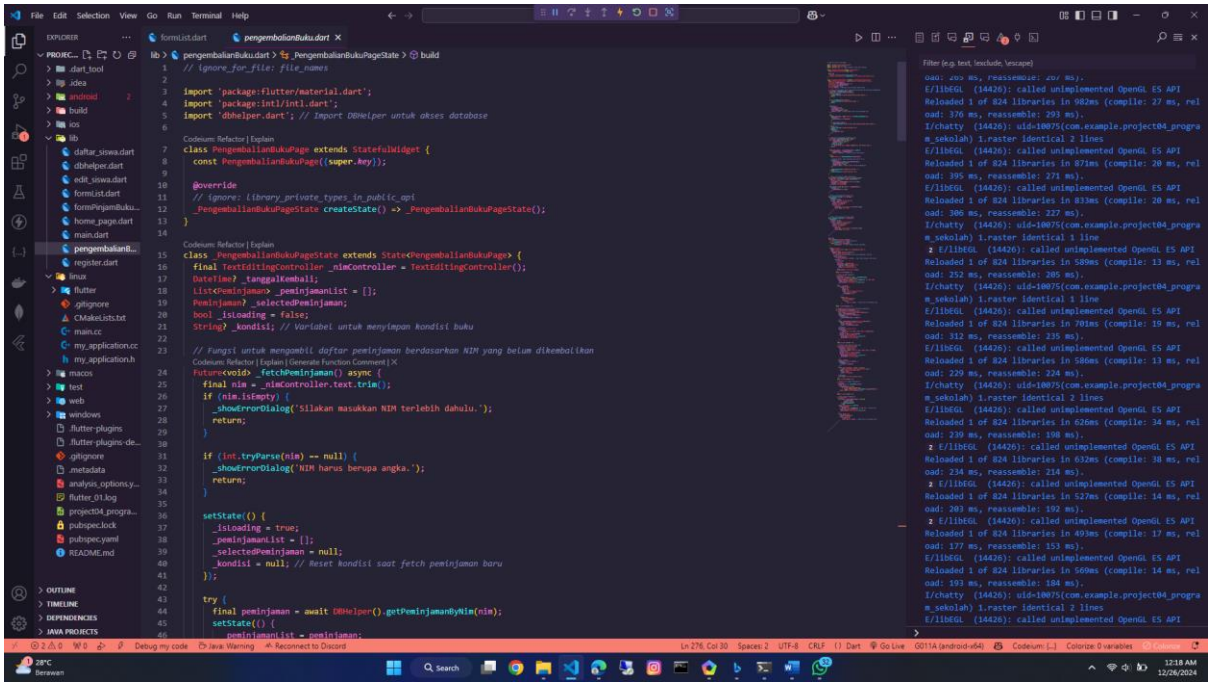
```dart
class PeminjamanPage extends StatefulWidget {
  const PeminjamanPage({super.key});

  @override
  // ignore: library_private_types_in_public_api
  _PeminjamanPageState createState() => _PeminjamanPageState();
}

class _PeminjamanPageState extends State<PeminjamanPage> {
  final TextEditingController _nimController = TextEditingController();
  final TextEditingController _bukuController = TextEditingController();
  final TextEditingController _jumlahController = TextEditingController();
  final TextEditingController _alasanController = TextEditingController();

  final List<String> _bukuList = [
    'Matematika Dasar',
    'Fisika',
    'Kimia Organik',
    'Sejarah Indonesia',
    'Biologi Modern',
    'Pemrograman Flutter'
  ];

  DateTime? _tanggalPinjam;
  DateTime? _tanggalKembali;

  Future<void> _selectDate(BuildContext context, bool isPinjam) async {
    final DateTime? picked = await showDatePicker(
      context: context,
      initialDate: isPinjam
          ? (_tanggalPinjam ?? DateTime.now())
          : (_tanggalKembali ?? DateTime.now().add(const Duration(days: 7))),
      firstDate: DateTime(2000),
      lastDate: DateTime(2100),
    );
    if (picked != null) {
      setState(() {
        if (isPinjam) {
          _tanggalPinjam = picked;
          if (_tanggalKembali != null &&
              _tanggalKembali!.isBefore(_tanggalPinjam!)) {
            _tanggalKembali = null;
          }
        } else {
          _tanggalKembali = picked;
```



```dart
              // Daftar Siswa Aktif
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                  minimumSize: const Size(200, 50),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(30),
                  ), // RoundedRectangleBorder
                  padding: const EdgeInsets.symmetric(vertical: 16),
                  textStyle: const TextStyle(
                      fontSize: 16, fontWeight: FontWeight.w600), // TextStyle
                ),
                onPressed: () {
                  Navigator.push(context, MaterialPageRoute(builder: (context) {
                    return const ListSiswaPage();
                  })); // MaterialPageRoute
                },
                child: const Text(
                  'Data Perpustakaan',
                  style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
                ), // Text
              ), // ElevatedButton
              const SizedBox(height: 20),

              // Daftar Siswa
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                  minimumSize: const Size(200, 50),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(30),
                  ), // RoundedRectangleBorder
                  padding: const EdgeInsets.symmetric(vertical: 16),
                  textStyle: const TextStyle(
                      fontSize: 16, fontWeight: FontWeight.w600), // TextStyle
                ),
                onPressed: () {
                  Navigator.push(context, MaterialPageRoute(builder: (context) {
                    return const DaftarSiswa();
                  })); // MaterialPageRoute
                },
                child: const Text(
                  'Data Siswa',
                  style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
                ), // Text
              ), // ElevatedButton
            ],
          ), // Column
        ), // Center
      ), // Padding
```



```dart
import 'package:flutter/material.dart';
import 'package:project04_program_sekolah/home_page.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: HomePage(),
      debugShowCheckedModeBanner: false,
    ); // MaterialApp
  }
}
```



```dart
// ignore_for_file: file_names

import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'dbhelper.dart'; // Import DBHelper untuk akses database

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

  @override
  // ignore: library_private_types_in_public_api
  _PengembalianBukuPageState createState() => _PengembalianBukuPageState();
}

class _PengembalianBukuPageState extends State<PengembalianBukuPage> {
  final TextEditingController _nimController = TextEditingController();
  DateTime? _tanggalKembali;
  List<Peminjaman> _peminjamanList = [];
  Peminjaman? _selectedPeminjaman;
  bool _isLoading = false;
  String? _kondisi; // Variabel untuk menyimpan kondisi buku

  // Fungsi untuk mengambil daftar peminjaman berdasarkan NIM yang belum dikembalikan
  Future<void> _fetchPeminjaman() async {
    final nim = _nimController.text.trim();
    if (nim.isEmpty) {
      _showErrorDialog('Silakan masukkan NIM terlebih dahulu.');
      return;
    }

    if (int.tryParse(nim) == null) {
      _showErrorDialog('NIM harus berupa angka.');
      return;
    }

    setState(() {
      _isLoading = true;
      _peminjamanList = [];
      _selectedPeminjaman = null;
      _kondisi = null; // Reset kondisi saat fetch peminjaman baru
    });

    try {
      final peminjaman = await DBHelper().getPeminjamanByNim(nim);
      setState(() {
        _peminjamanList = peminjaman;
```