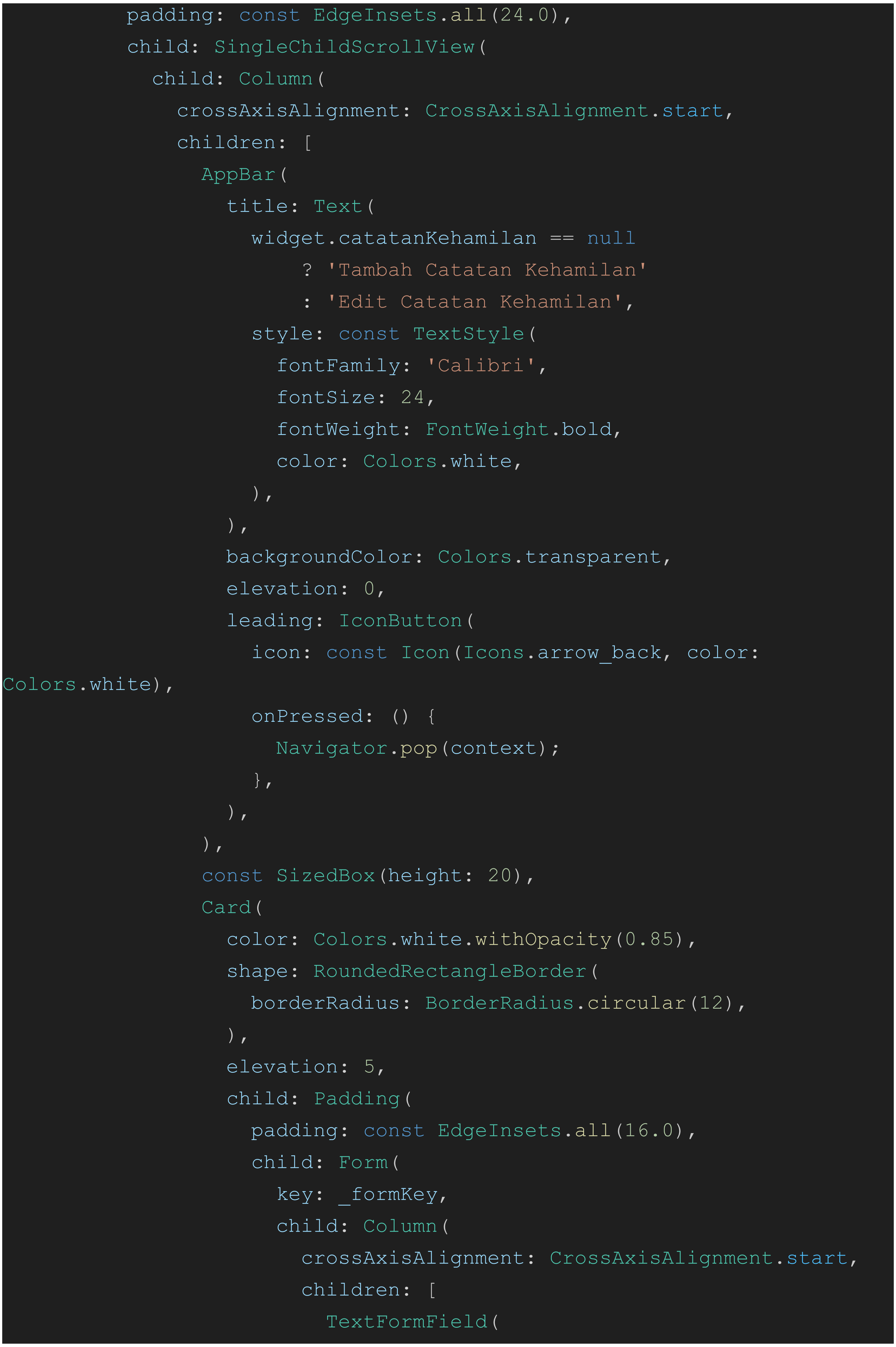
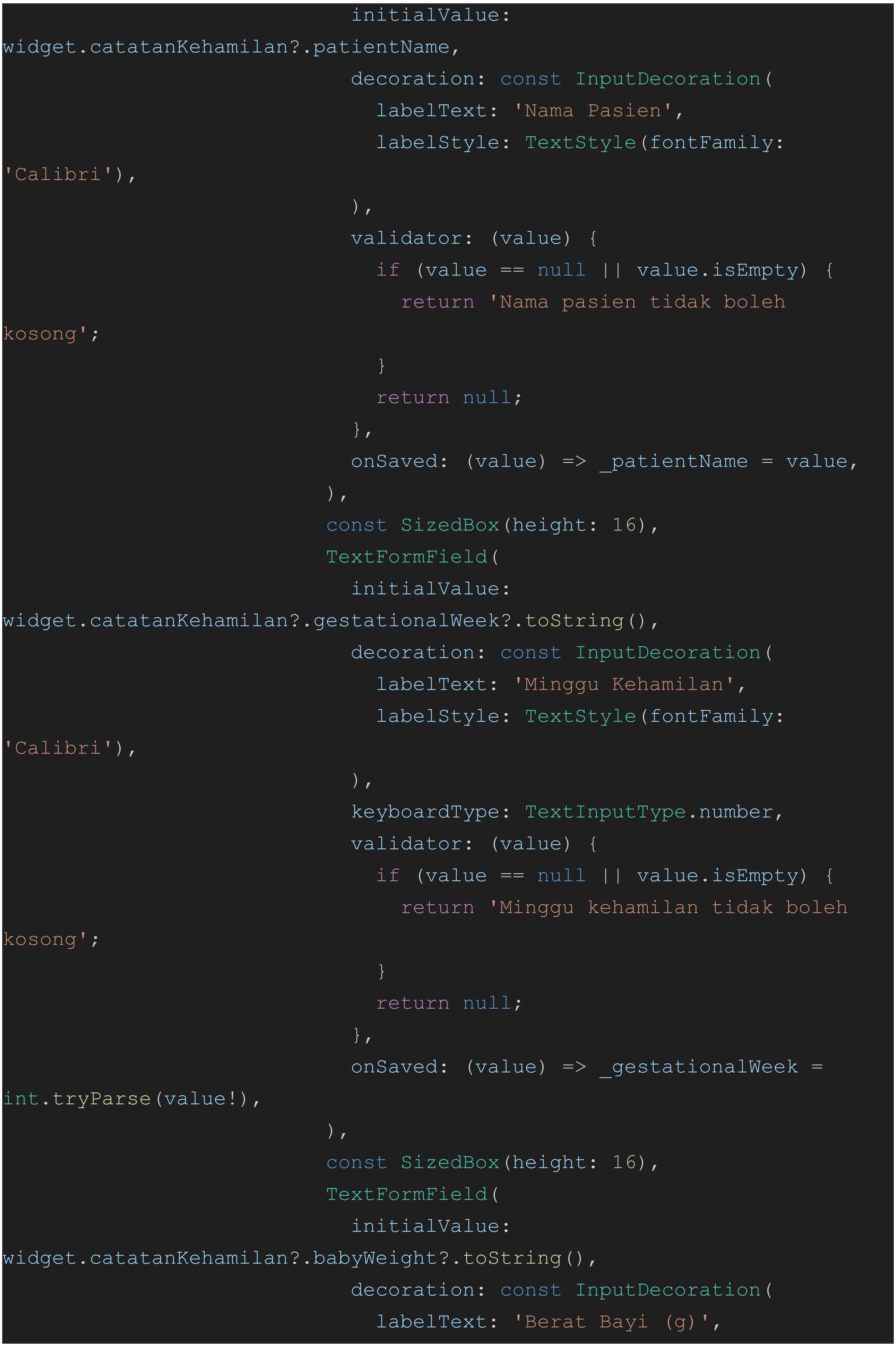
NAMA : HAMAS IZZUDDIN FATHI
NIM : H1D022097
MANAJEMEN CATATAN KEHAMILAN

CRUD
> Add

```
import 'package:flutter/material.dart';
import
'package:flutter_application_1/bloc/catatan_kehamilan_bloc.dart';
import 'package:flutter_application_1/model/kehamilan.dart';

class CatatanKehamilanForm extends StatefulWidget {
  final CatatanKehamilan? catatanKehamilan;

  CatatanKehamilanForm({this.catatanKehamilan});

  @override
  _CatatanKehamilanFormState createState() =>
_CatatanKehamilanFormState();
}

class _CatatanKehamilanFormState extends State<CatatanKehamilanForm> {
  final _formKey = GlobalKey<FormState>();
  String? _patientName;
  int? _gestationalWeek;
  int? _babyWeight;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        height: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Colors.orange[300]!,
              Colors.orange[600]!,
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Padding(
```

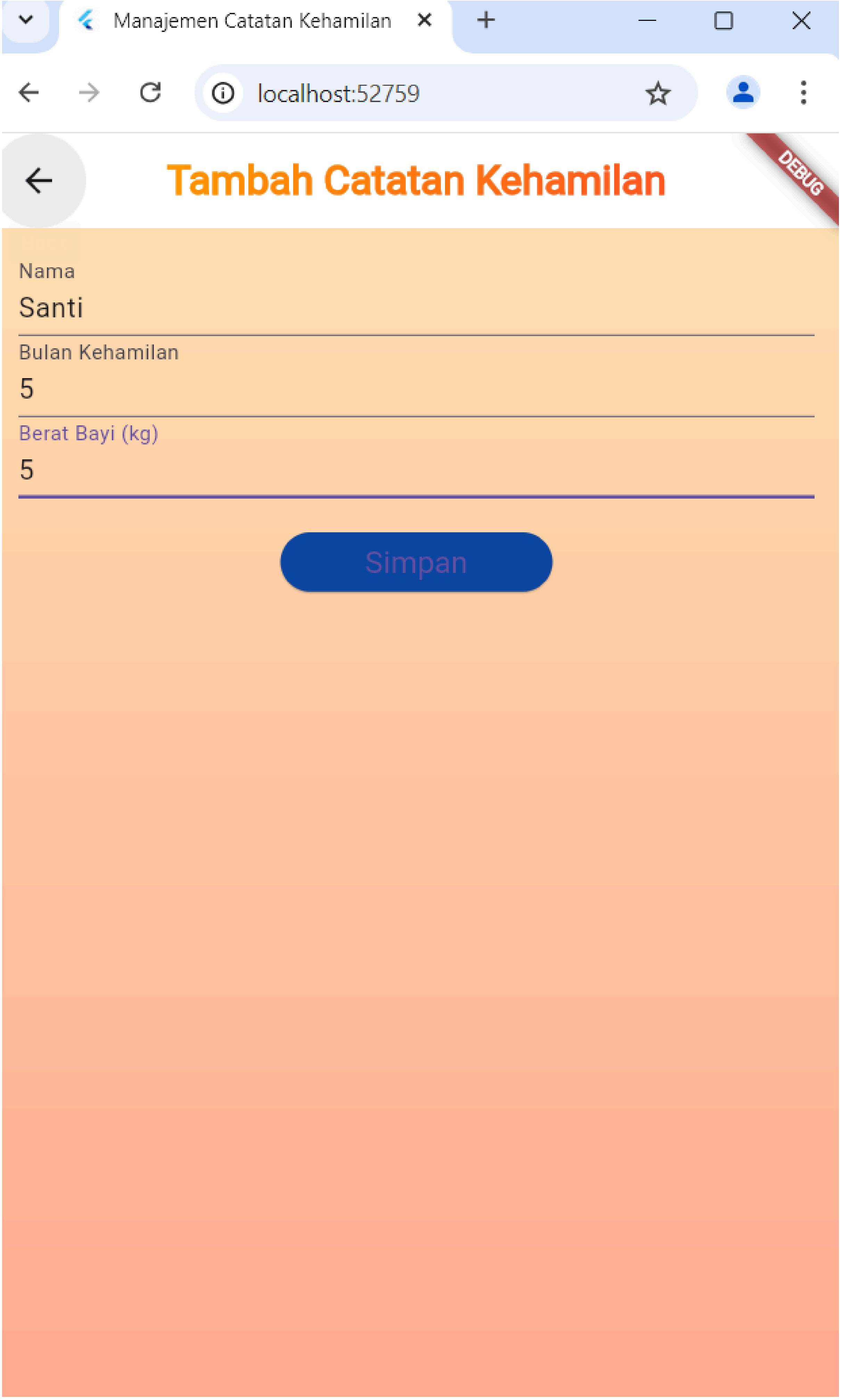


```
          padding: const EdgeInsets.all(24.0),
          child: SingleChildScrollView(
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                AppBar(
                  title: Text(
                    widget.catatanKehamilan == null
                        ? 'Tambah Catatan Kehamilan'
                        : 'Edit Catatan Kehamilan',
                    style: const TextStyle(
                      fontFamily: 'Calibri',
                      fontSize: 24,
                      fontWeight: FontWeight.bold,
                      color: Colors.white,
                    ),
                  ),
                  backgroundColor: Colors.transparent,
                  elevation: 0,
                  leading: IconButton(
                    icon: const Icon(Icons.arrow_back, color:
Colors.white),
                    onPressed: () {
                      Navigator.pop(context);
                    },
                  ),
                ),
                const SizedBox(height: 20),
                Card(
                  color: Colors.white.withOpacity(0.85),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                  ),
                  elevation: 5,
                  child: Padding(
                    padding: const EdgeInsets.all(16.0),
                    child: Form(
                      key: _formKey,
                      child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          TextFormField(
```



```
                            initialValue:
widget.catatanKehamilan?.patientName,
                            decoration: const InputDecoration(
                              labelText: 'Nama Pasien',
                              labelStyle: TextStyle(fontFamily:
'Calibri'),
                            ),
                            validator: (value) {
                              if (value == null || value.isEmpty) {
                                return 'Nama pasien tidak boleh
kosong';
                              }
                              return null;
                            },
                            onSaved: (value) => _patientName = value,
                          ),
                          const SizedBox(height: 16),
                          TextFormField(
                            initialValue:
widget.catatanKehamilan?.gestationalWeek?.toString(),
                            decoration: const InputDecoration(
                              labelText: 'Minggu Kehamilan',
                              labelStyle: TextStyle(fontFamily:
'Calibri'),
                            ),
                            keyboardType: TextInputType.number,
                            validator: (value) {
                              if (value == null || value.isEmpty) {
                                return 'Minggu kehamilan tidak boleh
kosong';
                              }
                              return null;
                            },
                            onSaved: (value) => _gestationalWeek =
int.tryParse(value!),
                          ),
                          const SizedBox(height: 16),
                          TextFormField(
                            initialValue:
widget.catatanKehamilan?.babyWeight?.toString(),
                            decoration: const InputDecoration(
                              labelText: 'Berat Bayi (g)',
                              labelStyle: TextStyle(fontFamily:
'Calibri'),
                            ),
                            keyboardType: TextInputType.number,
                            validator: (value) {
                              if (value == null || value.isEmpty) {
                                return 'Berat bayi tidak boleh kosong';
                              }
                              return null;
                            },
                            onSaved: (value) => _babyWeight =
int.tryParse(value!),
                          ),
                          const SizedBox(height: 20),
                          ElevatedButton(
                            style: ElevatedButton.styleFrom(
                              padding: const
EdgeInsets.symmetric(vertical: 15),
                              backgroundColor: Colors.blue[600], //
Warna tombol simpan
                              shape: RoundedRectangleBorder(
                                borderRadius:
BorderRadius.circular(10),
                              ),
                            ),
                            onPressed: () async {
                              if (_formKey.currentState!.validate()) {
                                _formKey.currentState!.save();

                                CatatanKehamilan newCatatanKehamilan =
CatatanKehamilan(
                                  id: widget.catatanKehamilan?.id, //
Menyimpan id jika ada
                                  patientName: _patientName!,
                                  gestationalWeek: _gestationalWeek,
                                  babyWeight: _babyWeight,
                                );

                                if (widget.catatanKehamilan == null) {
                                  // Menambahkan catatan kehamilan baru
                                  await
CatatanKehamilanBloc.addCatatanKehamilan(
                                      catatanKehamilan:
newCatatanKehamilan);
                                } else {
                                  // Memperbarui catatan kehamilan yang
ada
                                  await
CatatanKehamilanBloc.updateCatatanKehamilan(
                                      catatanKehamilan:
newCatatanKehamilan);
                                }

                                Navigator.pop(context);
                              }
                            },
                            child: const Text('Simpan', style:
TextStyle(fontFamily: 'Calibri')),
                          ),
                        ],
                      ),
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

- program diatas adalah crud add yang dimana akan menambah data pasien dengan format nama, bulan ke berapa, dan berat badan bayi
- dengan font calibri
- dengan background orange yaitu pastel



Tambah Catatan Kehamilan

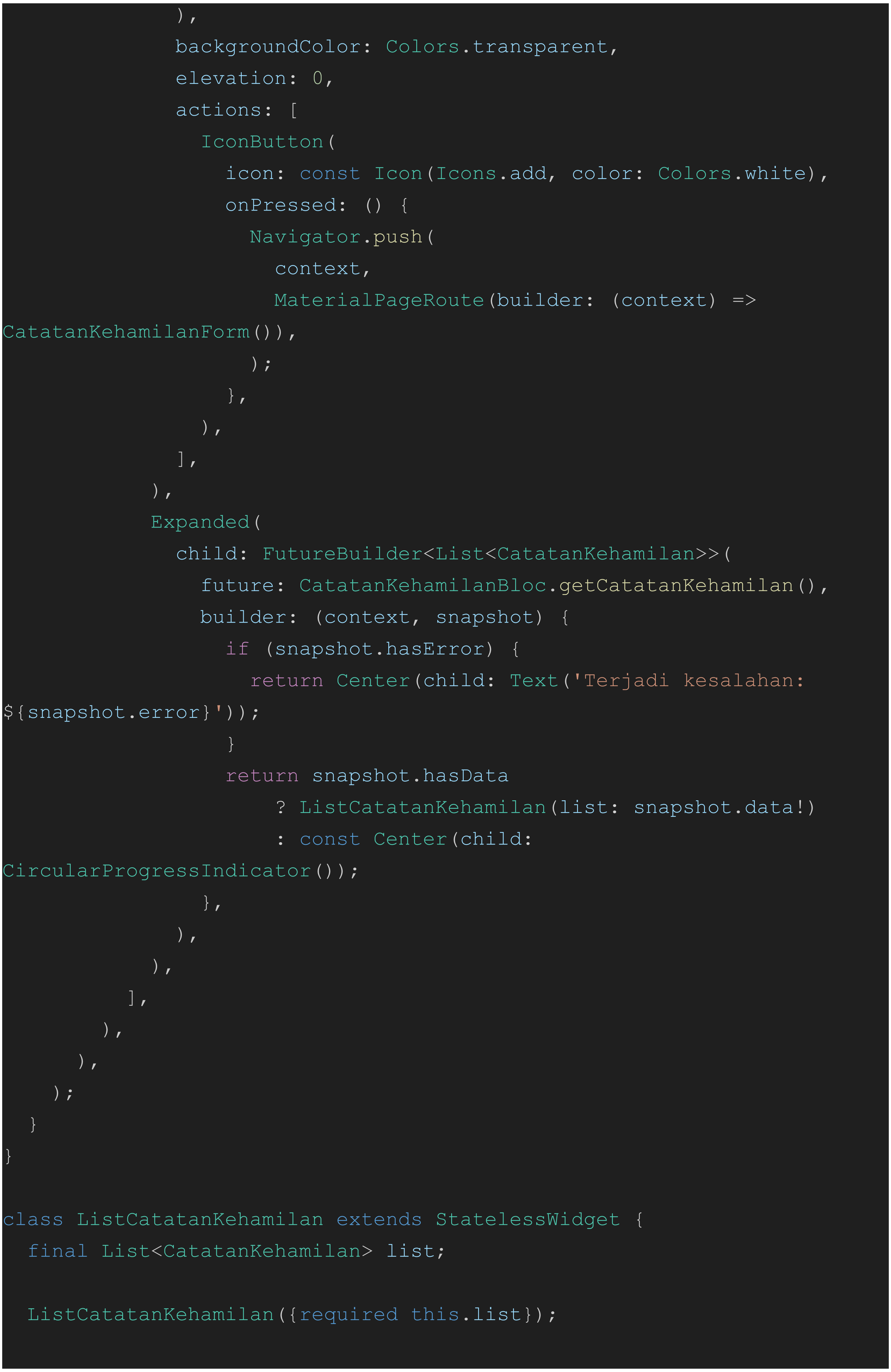
Nama
Santi

Bulan Kehamilan
5

Berat Bayi (kg)
5

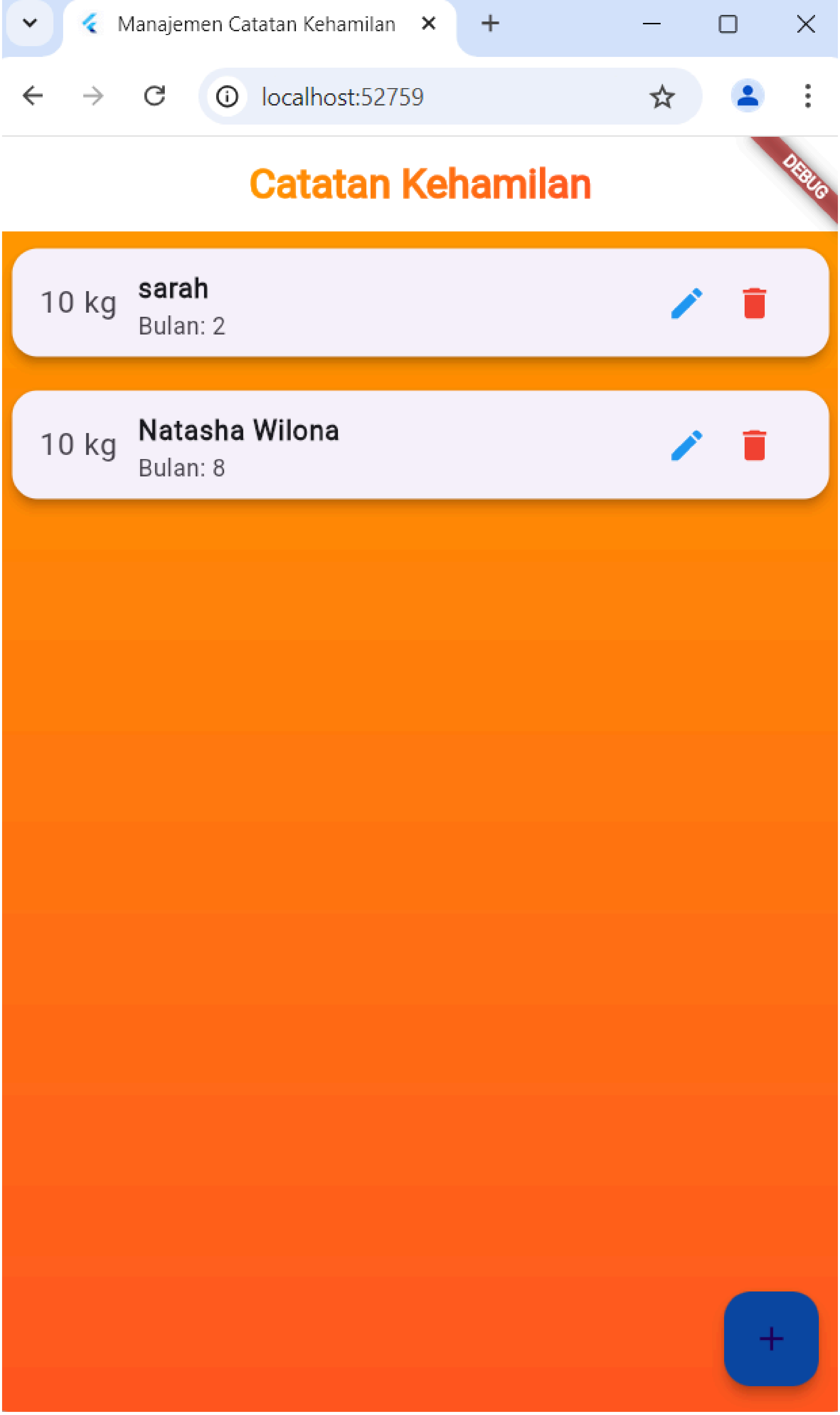
> kehamilan page

```
import 'package:flutter/material.dart';
import
'package:flutter_application_1/bloc/catatan_kehamilan_bloc.dart';
import 'package:flutter_application_1/model/kehamilan.dart';
import 'catatan_kehamilan_form.dart';
import 'catatan_kehamilan_detail.dart';

class CatatanKehamilanPage extends StatefulWidget {
  @override
  _CatatanKehamilanPageState createState() =>
_CatatanKehamilanPageState();
}

class _CatatanKehamilanPageState extends State<CatatanKehamilanPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        height: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Colors.orange[300]!,
              Colors.orange[600]!,
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Column(
          children: [
            AppBar(
              title: const Text(
                'Catatan Kehamilan',
                style: TextStyle(
                  fontFamily: 'Calibri',
                  fontSize: 24,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ),
```



```
              ),
              backgroundColor: Colors.transparent,
              elevation: 0,
              actions: [
                IconButton(
                  icon: const Icon(Icons.add, color: Colors.white),
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) =>
CatatanKehamilanForm()),
                    );
                  },
                ),
              ],
            ),
            Expanded(
              child: FutureBuilder<List<CatatanKehamilan>>(
                future: CatatanKehamilanBloc.getCatatanKehamilan(),
                builder: (context, snapshot) {
                  if (snapshot.hasError) {
                    return Center(child: Text('Terjadi kesalahan:
${snapshot.error}'));
                  }
                  return snapshot.hasData
                      ? ListCatatanKehamilan(list: snapshot.data!)
                      : const Center(child:
CircularProgressIndicator());
                },
              ),
            ),
          ],
        ),
      ),
    );
  }
}

class ListCatatanKehamilan extends StatelessWidget {
  final List<CatatanKehamilan> list;

  ListCatatanKehamilan({required this.list});
  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      padding: const EdgeInsets.all(8.0),
      itemCount: list.length,
      itemBuilder: (context, index) {
        final catatan = list[index];
        return Card(
          color: Colors.white.withOpacity(0.85), // Warna lebih terang
          elevation: 5,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12),
          ),
          child: ListTile(
            title: Text(
              'Nama Pasien: ${catatan.patientName}',
              style: const TextStyle(
                fontFamily: 'Calibri',
                fontSize: 18,
                fontWeight: FontWeight.bold,
                color: Colors.black87,
              ),
            ),
            subtitle: Text(
              'Minggu Kehamilan: ${catatan.gestationalWeek}, Berat
Bayi: ${catatan.babyWeight}g',
              style: const TextStyle(
                fontFamily: 'Calibri',
                fontSize: 16,
                color: Colors.black54,
              ),
            ),
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) =>
CatatanKehamilanDetail(catatanKehamilan: catatan),
                ),
              );
            },
          ),
        );
        },
      );
  }
}
```

- di kehamilan page ini berisi nama pasien yang telah ditambahkan dan jika di klik akan menuju tampilan detail
- di sini akan terdapat list pasien yang mendaftar di catatan kehamilan



Catatan Kehamilan

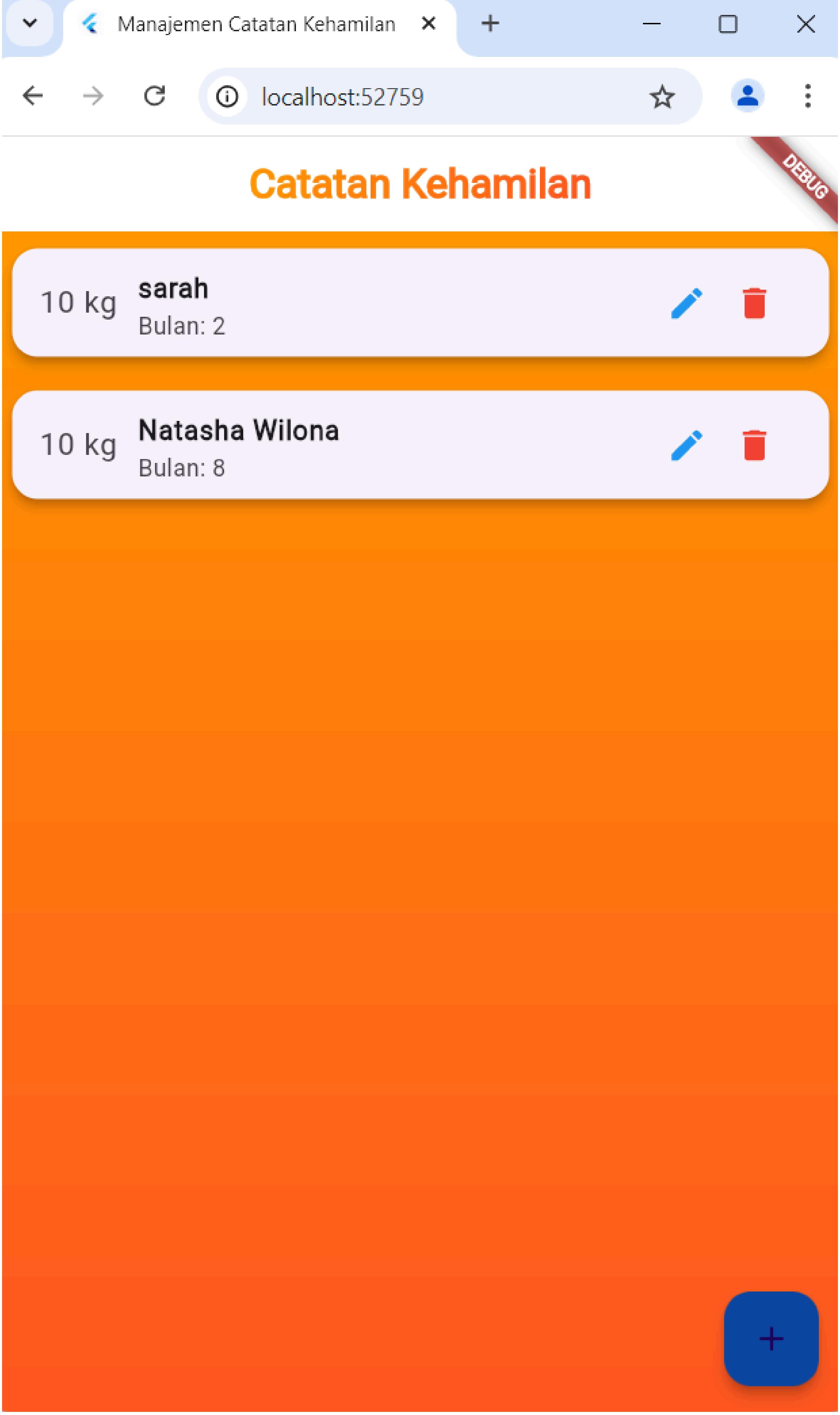
10 kg sarah
Bulan: 2

10 kg Natasha Wilona
Bulan: 8

> hapus dan ubah

```
import 'package:flutter/material.dart';
import
'package:flutter_application_1/bloc/catatan_kehamilan_bloc.dart';
import 'package:flutter_application_1/model/kehamilan.dart';
import 'catatan_kehamilan_form.dart';

class CatatanKehamilanDetail extends StatelessWidget {
  final CatatanKehamilan catatanKehamilan;

  CatatanKehamilanDetail({required this.catatanKehamilan});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        height: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Colors.orange[300]!,
              Colors.orange[600]!,
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                AppBar(
                  title: const Text(
                    'Detail Catatan Kehamilan',
                    style: TextStyle(
                      fontFamily: 'Calibri',
                      fontSize: 24,
                      fontWeight: FontWeight.bold,
                      color: Colors.white,
                    ),
                  ),
                  backgroundColor: Colors.transparent,
                  elevation: 0,
                  actions: [
                    IconButton(
                      icon: const Icon(Icons.edit, color:
Colors.white),
                      onPressed: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => CatatanKehamilanForm(
                              catatanKehamilan: catatanKehamilan,
                            ),
                          ),
                        );
                      },
                    ),
                  ],
                ),
                const SizedBox(height: 20),
                Card(
                  color: Colors.white.withOpacity(0.85),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                  ),
                  elevation: 5,
                  child: Padding(
                    padding: const EdgeInsets.all(16.0),
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Text(
                          'Nama Pasien:
${catatanKehamilan.patientName}',
                          style: const TextStyle(
                            fontFamily: 'Calibri',
                            fontSize: 18,
                            fontWeight: FontWeight.bold,
                            color: Colors.black87,
                          ),
                        ),
                        const SizedBox(height: 10),
                        Text(
                          'Minggu Kehamilan:
${catatanKehamilan.gestationalWeek}',
                          style: const TextStyle(
                            fontFamily: 'Calibri',
                            fontSize: 18,
                            color: Colors.black54,
                          ),
                        ),
                        const SizedBox(height: 10),
                        Text(
                          'Berat Bayi:
${catatanKehamilan.babyWeight}g',
                          style: const TextStyle(
                            fontFamily: 'Calibri',
                            fontSize: 18,
                            color: Colors.black54,
                          ),
                        ),
                        const SizedBox(height: 20),
                        ElevatedButton(
                          style: ElevatedButton.styleFrom(
                            padding: const
EdgeInsets.symmetric(vertical: 15),
                            backgroundColor: Colors.red[400], // Warna
tombol
                            shape: RoundedRectangleBorder(
                              borderRadius: BorderRadius.circular(10),
                            ),
                          ),
                          onPressed: () async {
                            // Menghapus catatan kehamilan
                            bool result = await
CatatanKehamilanBloc.deleteCatatanKehamilan(
                              id: catatanKehamilan.id as int,
                            );
                            if (result) {
                              Navigator.pop(context); // Kembali jika
berhasil dihapus
                            } else {
                              // Tampilkan pesan gagal
ScaffoldMessenger.of(context).showSnackBar(
                                SnackBar(
                                  content: Text('Gagal menghapus
catatan kehamilan'),
                                ),
                              );
                            }
                          },
                          child: const Text(
                            'Hapus',
                            style: TextStyle(
                              fontFamily: 'Calibri',
                              fontSize: 16,
                            ),
                          ),
                        ),
                      ],
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

- dengan diklik ubah maka akan bisa mengupdate yang telah kita tambah tadi
- dan jika di klik hapus maka akan hilang data tersebut



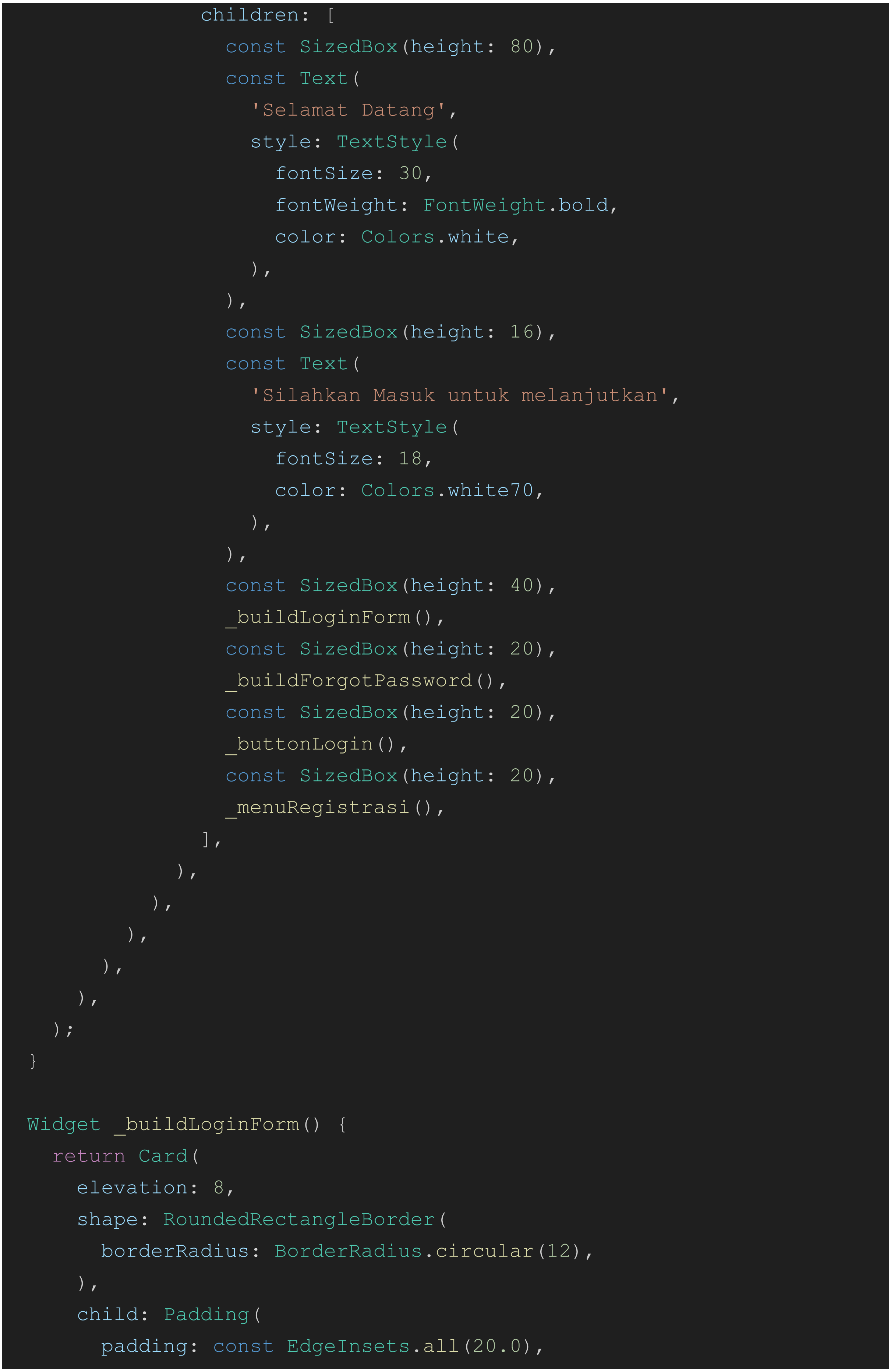
Catatan Kehamilan

10 kg sarah
Bulan: 2

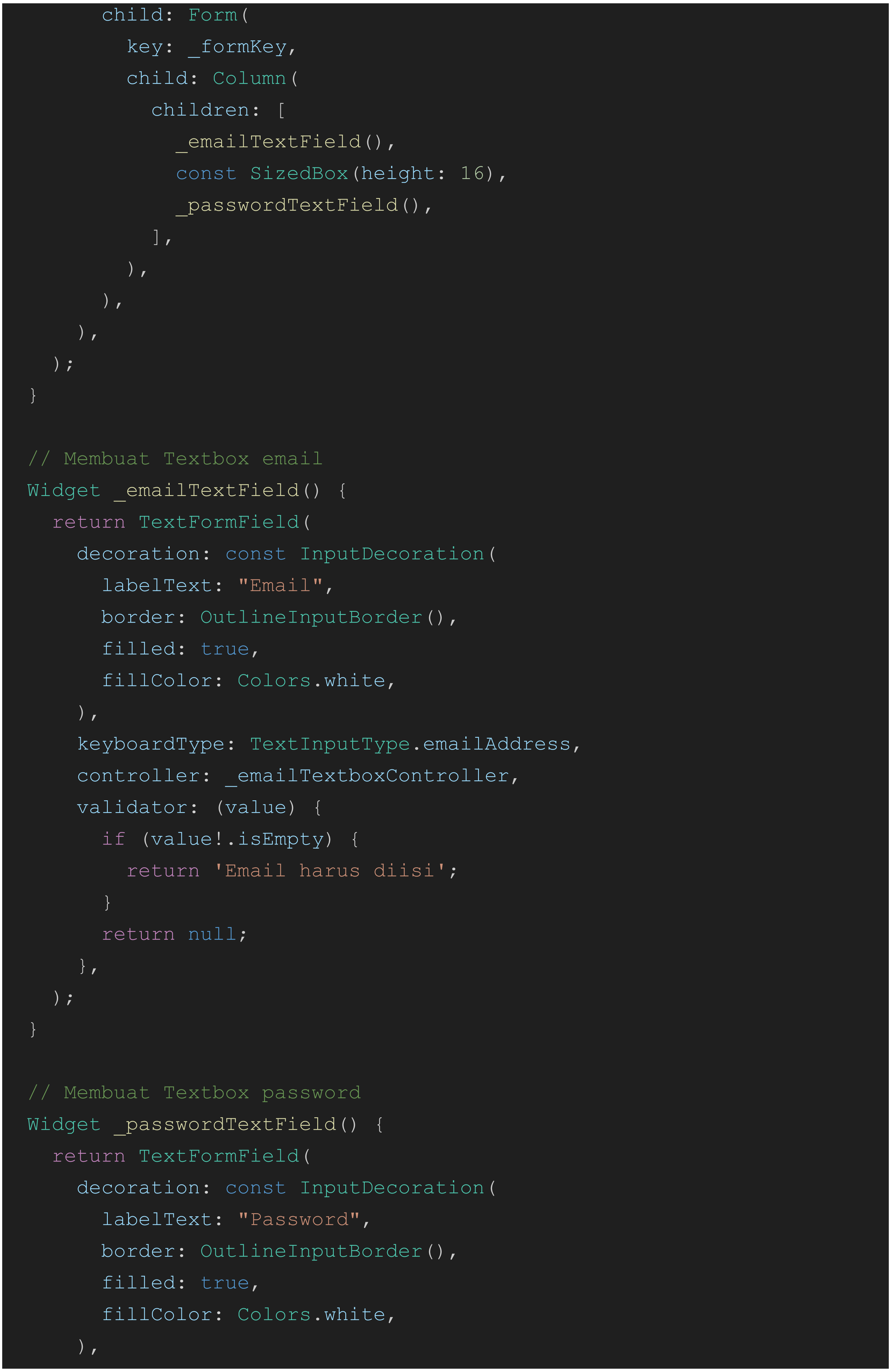
10 kg Natasha Wilona
Bulan: 8

> login

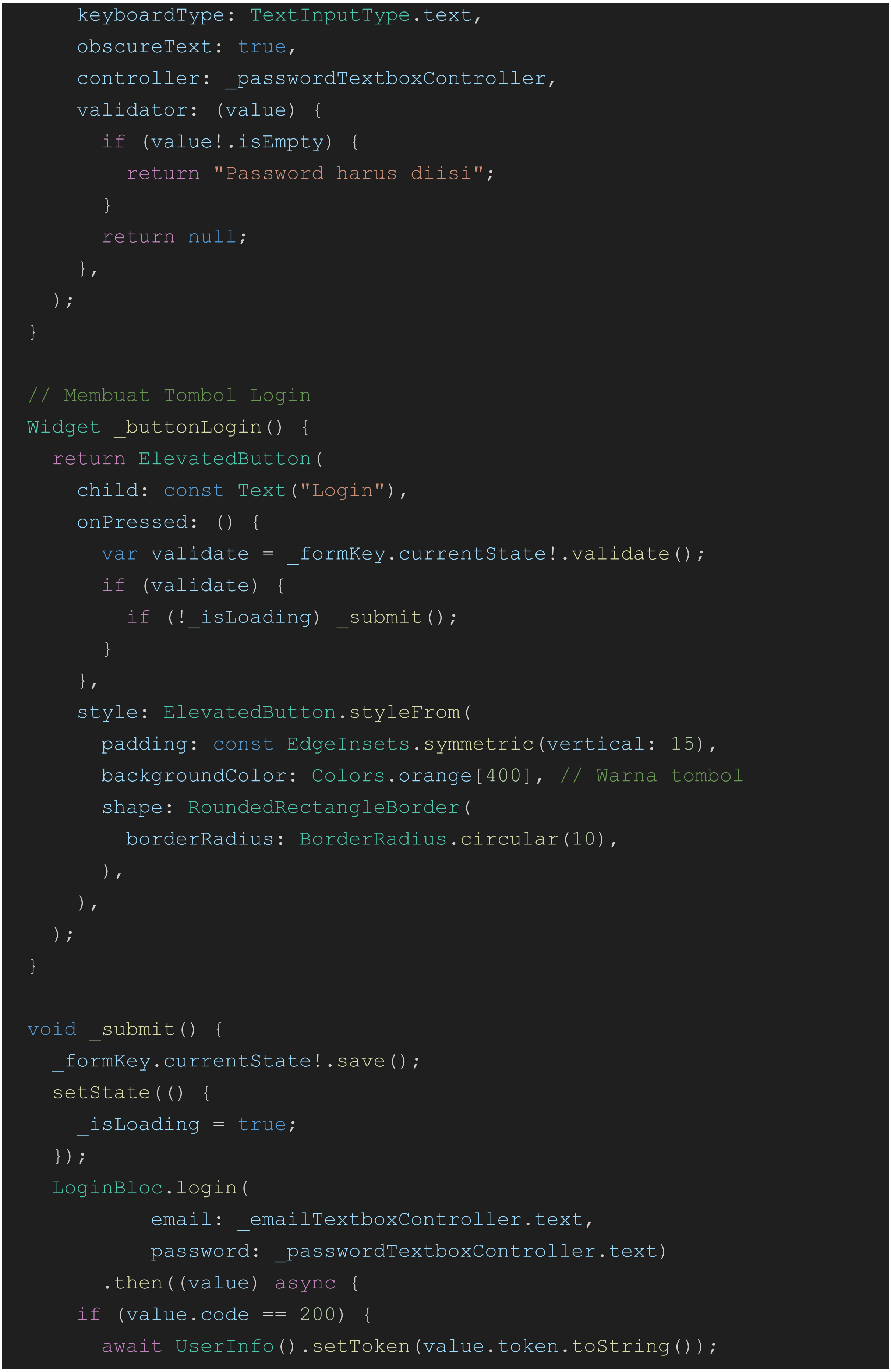
```
import 'package:flutter/material.dart';
import 'package:flutter_application_1/bloc/login_bloc.dart';
import 'package:flutter_application_1/helpers/user_info.dart';
import 'package:flutter_application_1/ui/catatan_kehamilan_page.dart';
import 'package:flutter_application_1/ui/registrasi_page.dart';
import 'package:flutter_application_1/widget/warning_dialog.dart';

class LoginPage extends StatefulWidget {
  const LoginPage({Key? key}) : super(key: key);

  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  final _emailTextboxController = TextEditingController();
  final _passwordTextboxController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        height: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Colors.orange[300]!,
              Colors.orange[600]!,
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(24.0),
            child: Center(
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
```



```
              children: [
                const SizedBox(height: 80),
                const Text(
                  'Selamat Datang',
                  style: TextStyle(
                    fontSize: 30,
                    fontWeight: FontWeight.bold,
                    color: Colors.white,
                  ),
                ),
                const SizedBox(height: 16),
                const Text(
                  'Silahkan Masuk untuk melanjutkan',
                  style: TextStyle(
                    fontSize: 18,
                    color: Colors.white70,
                  ),
                ),
                const SizedBox(height: 40),
                _buildLoginForm(),
                const SizedBox(height: 20),
                _buildForgotPassword(),
                const SizedBox(height: 20),
                _buttonLogin(),
                const SizedBox(height: 20),
                _menuRegistrasi(),
              ],
            ),
          ),
        ),
      ),
    ),
  );
}

Widget _buildLoginForm() {
  return Card(
    elevation: 8,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.all(20.0),
```



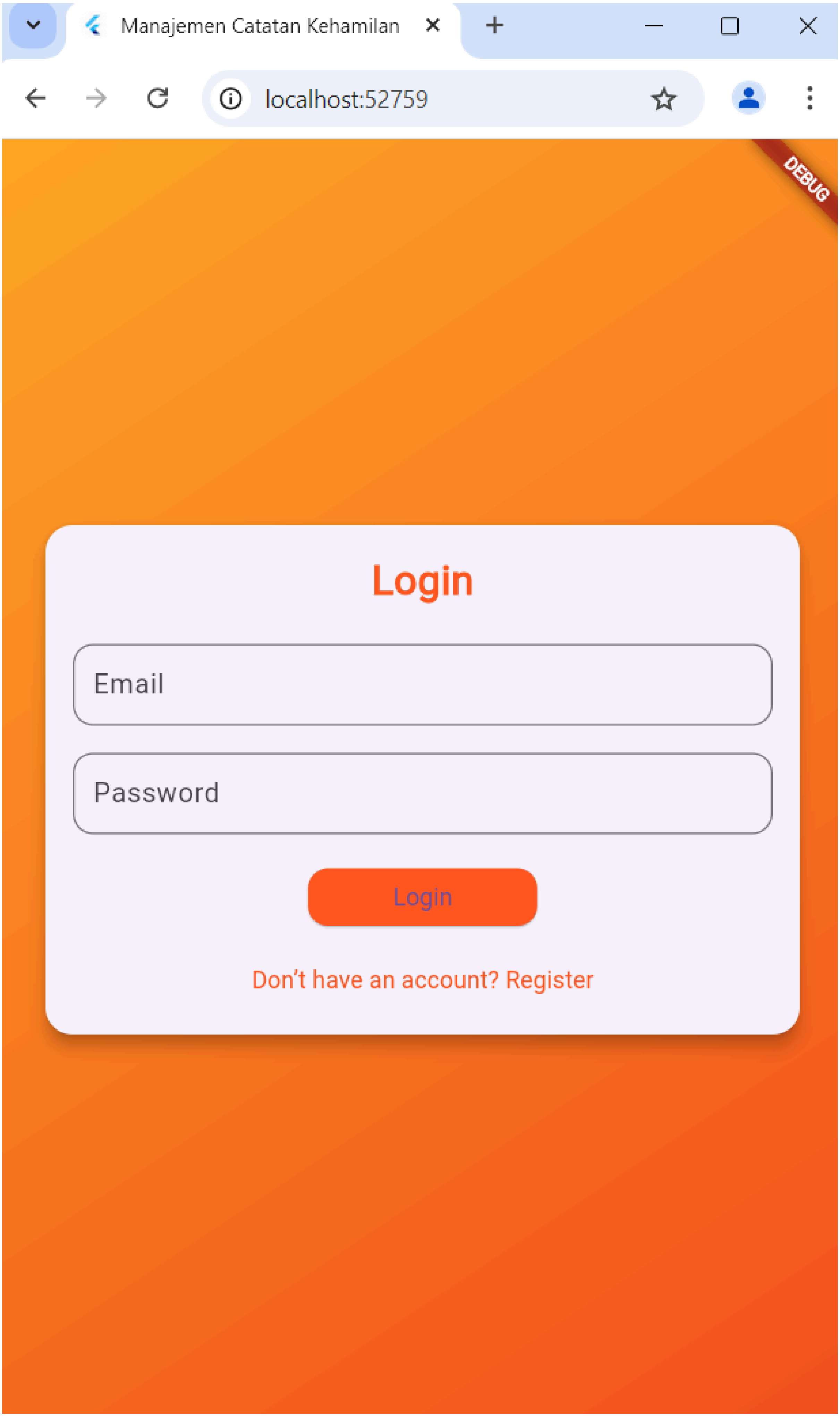
```
      child: Form(
        key: _formKey,
        child: Column(
          children: [
            _emailTextField(),
            const SizedBox(height: 16),
            _passwordTextField(),
          ],
        ),
      ),
    ),
  );
}

// Membuat Textbox email
Widget _emailTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Email",
      border: OutlineInputBorder(),
      filled: true,
      fillColor: Colors.white,
    ),
    keyboardType: TextInputType.emailAddress,
    controller: _emailTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return 'Email harus diisi';
      }
      return null;
    },
  );
}

// Membuat Textbox password
Widget _passwordTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Password",
      border: OutlineInputBorder(),
      filled: true,
      fillColor: Colors.white,
    ),
```



```
    keyboardType: TextInputType.text,
    obscureText: true,
    controller: _passwordTextboxController,
    validator: (value) {
      if (value!.isEmpty) {
        return "Password harus diisi";
      }
      return null;
    },
  );
}

// Membuat Tombol Login
Widget _buttonLogin() {
  return ElevatedButton(
    child: const Text("Login"),
    onPressed: () {
      var validate = _formKey.currentState!.validate();
      if (validate) {
        if (!_isLoading) _submit();
      }
    },
    style: ElevatedButton.styleFrom(
      padding: const EdgeInsets.symmetric(vertical: 15),
      backgroundColor: Colors.orange[400], // Warna tombol
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
    ),
  );
}

void _submit() {
  _formKey.currentState!.save();
  setState(() {
    _isLoading = true;
  });
  LoginBloc.login(
          email: _emailTextboxController.text,
          password: _passwordTextboxController.text)
      .then((value) async {
    if (value.code == 200) {
      await UserInfo().setToken(value.token.toString());
```

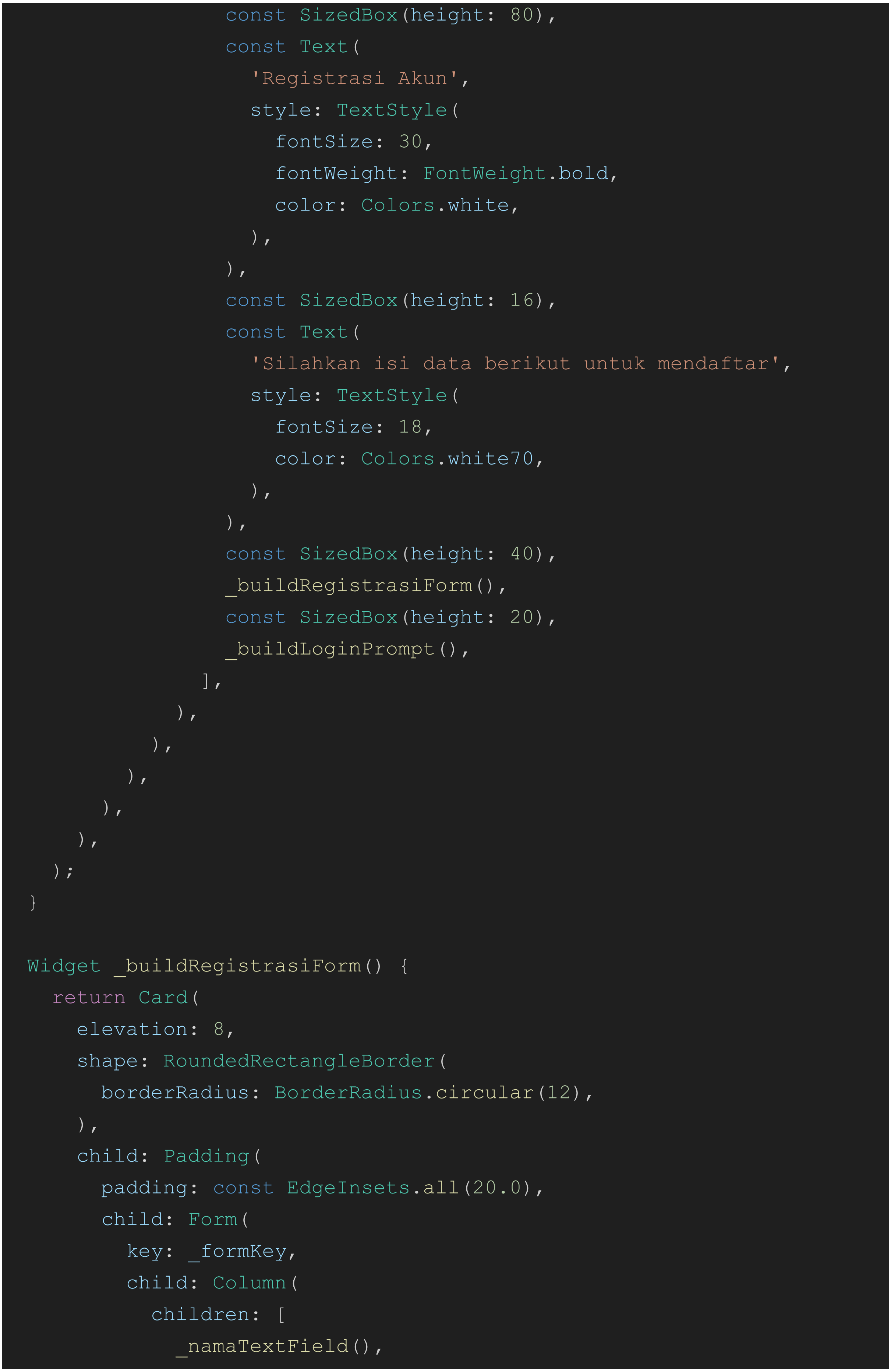
```
        await UserInfo().setUserID(value.userID as int);
        Navigator.pushReplacement(context,
            MaterialPageRoute(builder: (context) =>
CatatanKehamilanPage()));
      } else {
        showDialog(
            context: context,
            barrierDismissible: false,
            builder: (BuildContext context) => const WarningDialog(
                  description: "Login gagal, silahkan coba lagi",
                ));
      }
    }, onError: (error) {
      print(error);
      showDialog(
          context: context,
          barrierDismissible: false,
          builder: (BuildContext context) => const WarningDialog(
                description: "Login gagal, silahkan coba lagi",
              ));
    });
    setState(() {
      _isLoading = false;
    });
  }

  // Membuat menu untuk membuka halaman registrasi
  Widget _menuRegistrasi() {
    return Center(
      child: InkWell(
        child: const Text(
          "Belum punya akun? Daftar sekarang",
          style: TextStyle(color: Colors.white),
        ),
        onTap: () {
          Navigator.push(context,
              MaterialPageRoute(builder: (context) => const
RegistrasiPage()));
        },
      ),
    );
  }
  Widget _buildForgotPassword() {
    return Align(
      alignment: Alignment.centerRight,
      child: TextButton(
        onPressed: () {
          // Tindakan untuk lupa password
        },
        child: const Text(
          "Lupa Password?",
          style: TextStyle(color: Colors.white),
        ),
      ),
    );
  }
}
```

- di login ini anda harus regis terlebih dahulu untuk bisa login dan akan mengisi email dan password dengan api yang sudah di ambil oleh register

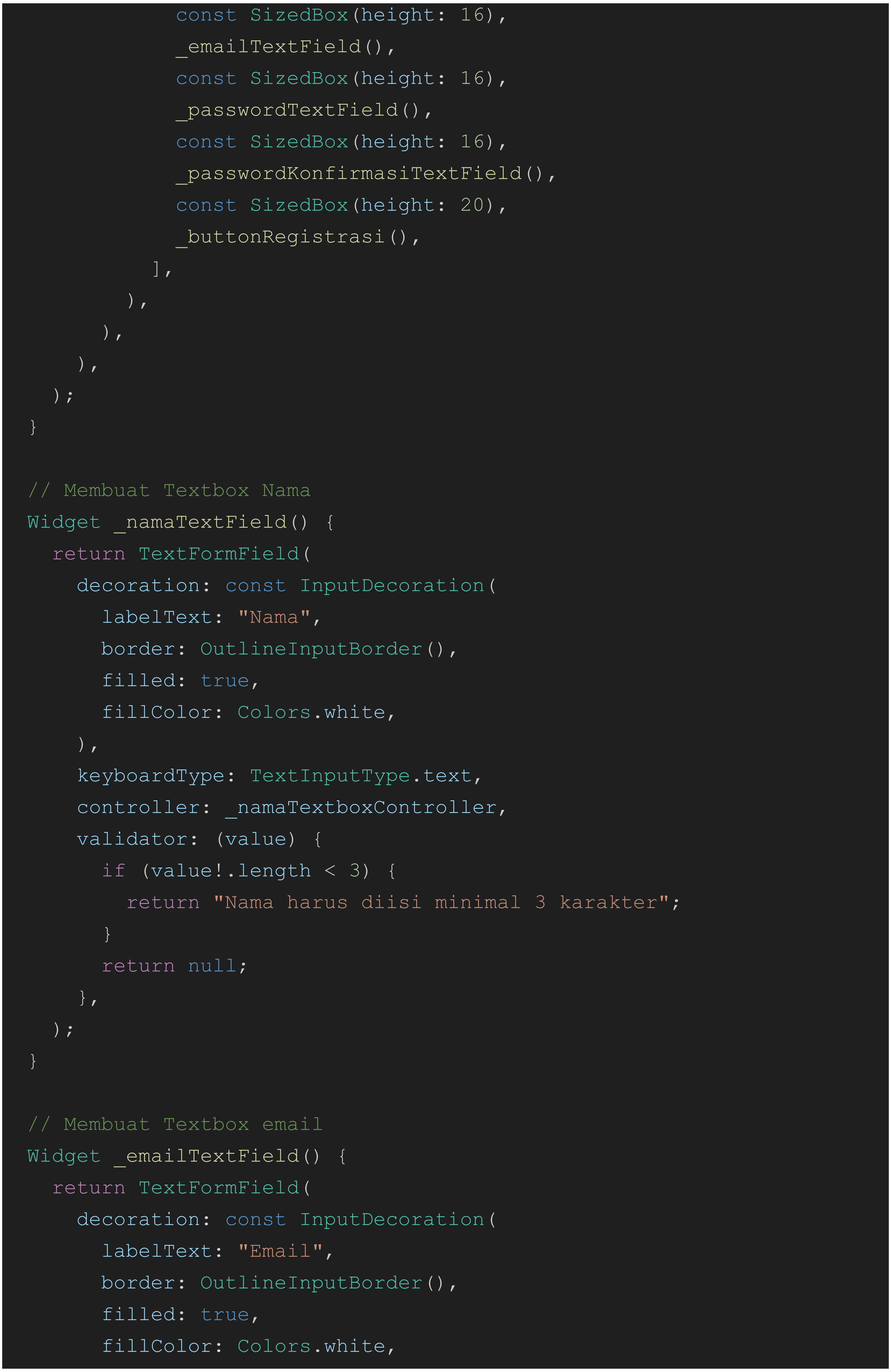
Manajemen Catatan Kehamilan
localhost:52759
DEBUG
Login
Email
Password
Login
Don’t have an account? Register

>register

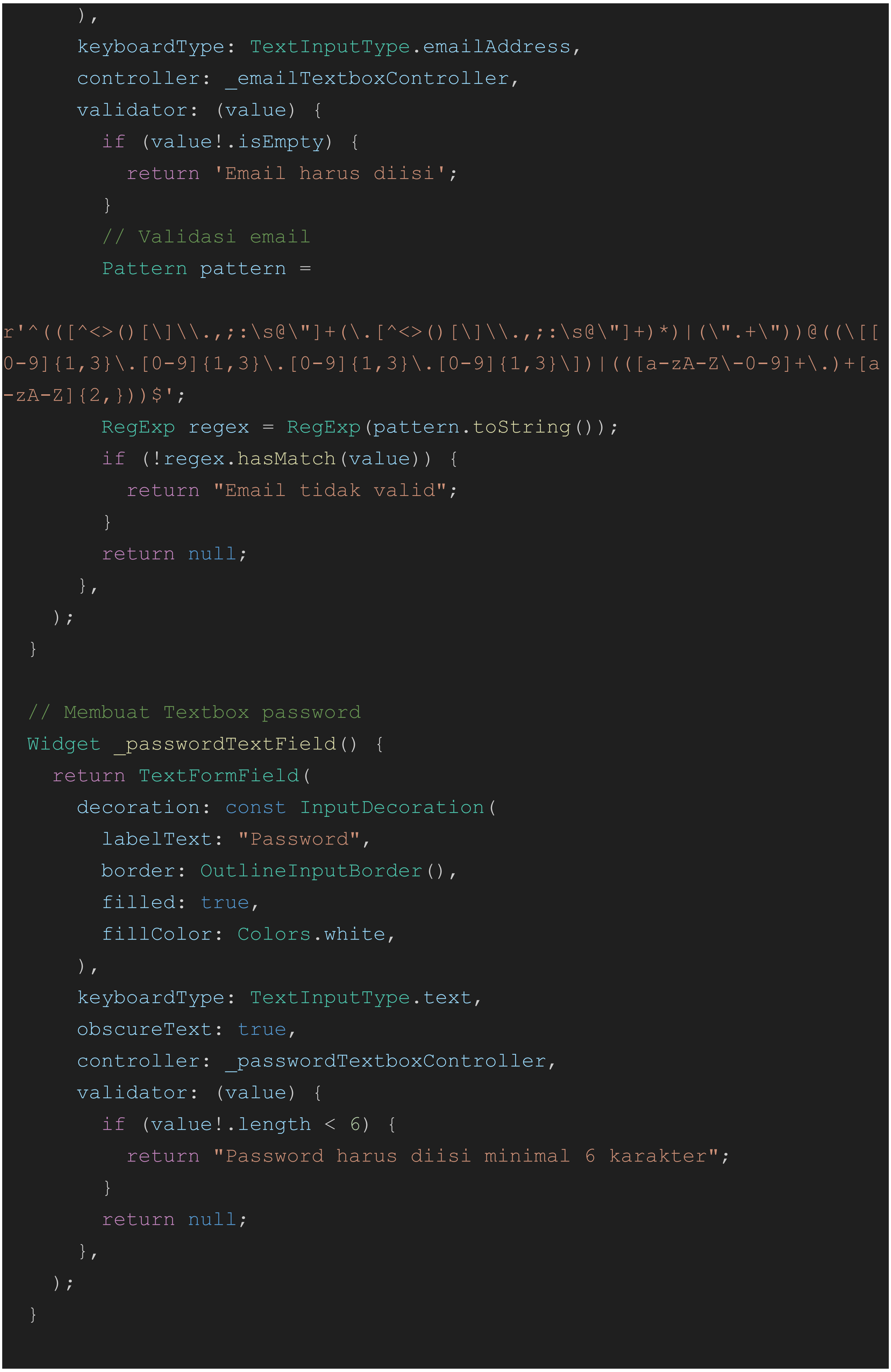
```dart
import 'package:flutter/material.dart';
import 'package:flutter_application_1/bloc/registrasi_bloc.dart';
import 'package:flutter_application_1/widget/succes_dialog.dart';
import 'package:flutter_application_1/widget/warning_dialog.dart';

class RegistrasiPage extends StatefulWidget {
  const RegistrasiPage({Key? key}) : super(key: key);

  @override
  _RegistrasiPageState createState() => _RegistrasiPageState();
}

class _RegistrasiPageState extends State<RegistrasiPage> {
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  final _namaTextboxController = TextEditingController();
  final _emailTextboxController = TextEditingController();
  final _passwordTextboxController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        height: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Colors.orange[300]!,
              Colors.orange[600]!,
            ],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(24.0),
            child: Center(
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
```



```
                const SizedBox(height: 80),
                const Text(
                  'Registrasi Akun',
                  style: TextStyle(
                    fontSize: 30,
                    fontWeight: FontWeight.bold,
                    color: Colors.white,
                  ),
                ),
                const SizedBox(height: 16),
                const Text(
                  'Silahkan isi data berikut untuk mendaftar',
                  style: TextStyle(
                    fontSize: 18,
                    color: Colors.white70,
                  ),
                ),
                const SizedBox(height: 40),
                _buildRegistrasiForm(),
                const SizedBox(height: 20),
                _buildLoginPrompt(),
              ],
            ),
          ),
        ),
      ),
    ),
  );
}

Widget _buildRegistrasiForm() {
  return Card(
    elevation: 8,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.all(20.0),
      child: Form(
        key: _formKey,
        child: Column(
          children: [
            _namaTextField(),
```

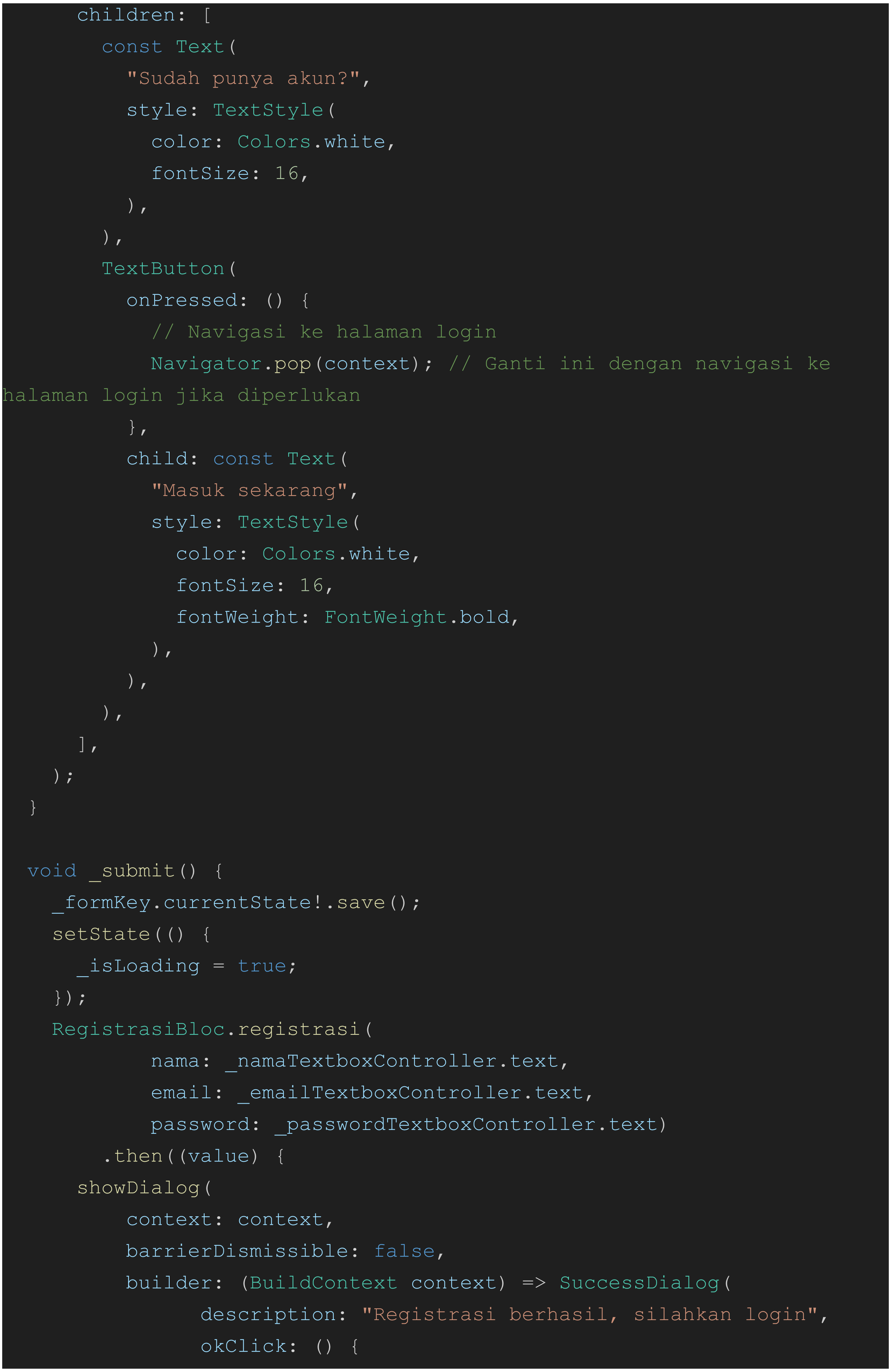


```
            const SizedBox(height: 16),
            _emailTextField(),
            const SizedBox(height: 16),
            _passwordTextField(),
            const SizedBox(height: 16),
            _passwordKonfirmasiTextField(),
            const SizedBox(height: 20),
            _buttonRegistrasi(),
          ],
        ),
      ),
    ),
  );
}

// Membuat Textbox Nama
Widget _namaTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Nama",
      border: OutlineInputBorder(),
      filled: true,
      fillColor: Colors.white,
    ),
    keyboardType: TextInputType.text,
    controller: _namaTextboxController,
    validator: (value) {
      if (value!.length < 3) {
        return "Nama harus diisi minimal 3 karakter";
      }
      return null;
    },
  );
}

// Membuat Textbox email
Widget _emailTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Email",
      border: OutlineInputBorder(),
      filled: true,
      fillColor: Colors.white,
```



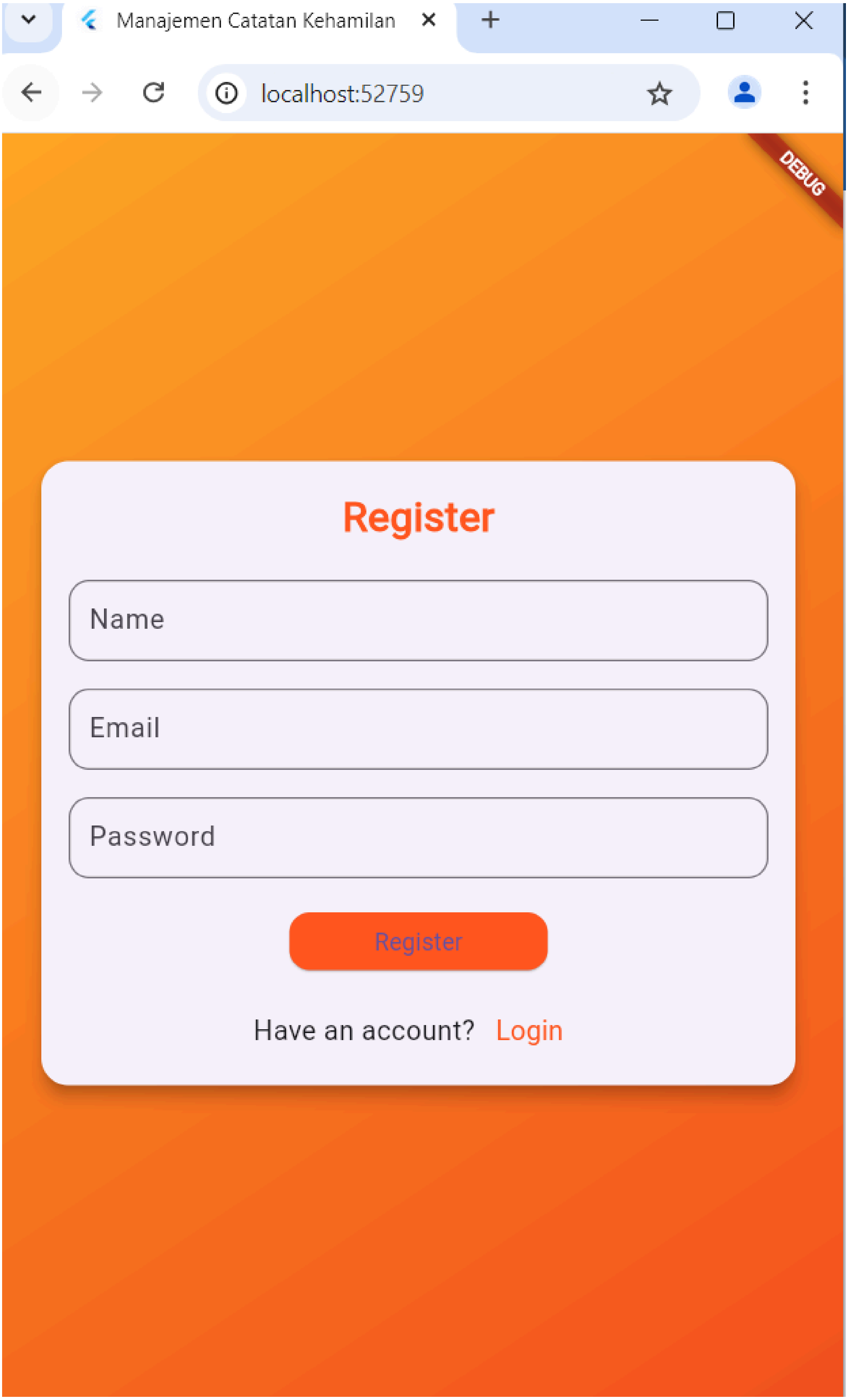
```
      ),
      keyboardType: TextInputType.emailAddress,
      controller: _emailTextboxController,
      validator: (value) {
        if (value!.isEmpty) {
          return 'Email harus diisi';
        }
        // Validasi email
        Pattern pattern =

r'^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[
0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a
-zA-Z]{2,}))$';
        RegExp regex = RegExp(pattern.toString());
        if (!regex.hasMatch(value)) {
          return "Email tidak valid";
        }
        return null;
      },
    );
  }

  // Membuat Textbox password
  Widget _passwordTextField() {
    return TextFormField(
      decoration: const InputDecoration(
        labelText: "Password",
        border: OutlineInputBorder(),
        filled: true,
        fillColor: Colors.white,
      ),
      keyboardType: TextInputType.text,
      obscureText: true,
      controller: _passwordTextboxController,
      validator: (value) {
        if (value!.length < 6) {
          return "Password harus diisi minimal 6 karakter";
        }
        return null;
      },
    );
  }
```

```
// Membuat textbox Konfirmasi Password
Widget _passwordKonfirmasiTextField() {
  return TextFormField(
    decoration: const InputDecoration(
      labelText: "Konfirmasi Password",
      border: OutlineInputBorder(),
      filled: true,
      fillColor: Colors.white,
    ),
    keyboardType: TextInputType.text,
    obscureText: true,
    validator: (value) {
      if (value != _passwordTextboxController.text) {
        return "Konfirmasi Password tidak sama";
      }
      return null;
    },
  );
}

// Membuat Tombol Registrasi
Widget _buttonRegistrasi() {
  return ElevatedButton(
    child: const Text("Registrasi"),
    onPressed: () {
      var validate = _formKey.currentState!.validate();
      if (validate) {
        if (!_isLoading) _submit();
      }
    },
    style: ElevatedButton.styleFrom(
      padding: const EdgeInsets.symmetric(vertical: 15),
      backgroundColor: Colors.orange[400], // Warna tombol
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(15),
      ),
    ),
  );
}

// Menambahkan teks untuk prompt login
Widget _buildLoginPrompt() {
  return Column(
```



```
      children: [
        const Text(
          "Sudah punya akun?",
          style: TextStyle(
            color: Colors.white,
            fontSize: 16,
          ),
        ),
        TextButton(
          onPressed: () {
            // Navigasi ke halaman login
            Navigator.pop(context); // Ganti ini dengan navigasi ke
halaman login jika diperlukan
          },
          child: const Text(
            "Masuk sekarang",
            style: TextStyle(
              color: Colors.white,
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ],
    );
  }

  void _submit() {
    _formKey.currentState!.save();
    setState(() {
      _isLoading = true;
    });
    RegistrasiBloc.registrasi(
            nama: _namaTextboxController.text,
            email: _emailTextboxController.text,
            password: _passwordTextboxController.text)
        .then((value) {
      showDialog(
          context: context,
          barrierDismissible: false,
          builder: (BuildContext context) => SuccessDialog(
                description: "Registrasi berhasil, silahkan login",
                okClick: () {
                  Navigator.pop(context);
                },
              ));
    }, onError: (error) {
      showDialog(
          context: context,
          barrierDismissible: false,
          builder: (BuildContext context) => const WarningDialog(
                description: "Registrasi gagal, silahkan coba lagi",
              ));
    });
    setState(() {
      _isLoading = false;
    });
  }
}
```

- register ini akan membuat user baru yang akan nanti dimasukkan ke login
- dan ini membutuhkan api yang sudah di buat di atas



Have an account? Login