

Nama : Juan Fakhri
NIM : 311910361
Kelas : TI.19.D5

MEMBUAT DATA KONTAK DENGAN FLUTTER DAN SQLITE BAGIAN 2

Assalamualaikum Warahmatullahi Wabarakatuh.

proses CRUD(create,read,update,delete) untuk database tabel kontak pada pertemuan sebelumnya:

```
import 'package:flutter/material.dart';
import 'package:mobcontact/databases/db_helper.dart';
import 'package:mobcontact/models/contact_model.dart';
import 'package:mobcontact/pages/form_contact.dart';

class ListContact extends StatefulWidget {
  const ListContact({Key? key}) : super(key: key);
  @override
  State<ListContact> createState() => _ListContactState();
}

class _ListContactState extends State<ListContact> {
  final List<Contact> listContact = [];
  final DbHelper dbHelper = DbHelper();
  Future<List<Contact>> getAllContact() async {
    listContact.clear();
    await dbHelper.getListContact().then((value) {
      (value as List).forEach((json) {
        listContact.add(Contact.fromJson(json));
      });
    });
    return listContact;
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('List Contact'))
    );
  }
}
```

```
import 'package:flutter/material.dart';
import 'package:mobcontact/databases/db_helper.dart';
import 'package:mobcontact/models/contact_model.dart';
import 'package:mobcontact/pages/form_contact.dart';

class ListContact extends StatefulWidget {
  const ListContact({Key? key}) : super(key: key);
  @override
  State<ListContact> createState() => _ListContactState();
}
```

```

}
class _ListContactState extends State<ListContact> {
  final List<Contact> listContact = [];
  final DbHelper dbHelper = DbHelper();
  Future<List<Contact>> getAllContact() async {
    listContact.clear();
    await dbHelper.getListContact().then((value) {
      (value as List).forEach((json) {
        listContact.add(Contact.fromJson(json));
      });
    });
    return listContact;
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("List Contacts")),
      body: FutureBuilder<List<Contact>>(
        future: getAllContact(),
        builder:
          (BuildContext context, AsyncSnapshot<List<Contact>>
            snapshot) {
            if (snapshot.hasData) {
              List<Contact> contacts = snapshot.data!;
              return ListView.builder(
                itemCount: contacts.length,
                itemBuilder: (context, index) {
                  Contact contact = contacts[index];
                  return ListTile(
                    leading: const CircleAvatar(
                      radius: 25,
                      backgroundColor: Colors.amber,
                    ),
                    title: Text("${contact.name}"),
                    subtitle: Text("${contact.email},
                      ${contact.mobileNo}"),
                    trailing: FittedBox(
                      fit: BoxFit.fill,
                      child: Row(
                        children: [
                          IconButton(
                            onPressed: () async {
                              await
                                dbHelper.delete(contact.id!);
                              setState(() {});

```

```

},
icon: const Icon(Icons.delete)),
IconButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          FormContact(contact)));
  },
  icon: const Icon(Icons.edit)),
],
)));
});
} else {
  return const Center(
    child: Text("Belum ada data"),
  );
}
}),
floatingActionButton: Row(
  mainAxisAlignment: MainAxisAlignment.spaceAround,
  children: [
    FloatingActionButton(
      heroTag: "add contact",
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => FormContact(Contact())));
      },
      child: const Icon(Icons.add),
    ),
    FloatingActionButton(
      onPressed: () async {
        getAllContact();
      },
      child: const Icon(Icons.refresh),
    ),
  ],
),
);
}
}

```

Form_contact.dart

```
form_contact.dart X pubspec.yaml
mobcontact > lib > pages > form_contact.dart > _FormContactState > build

1  import 'dart:io';
2  import 'package:flutter/material.dart';
3  import 'package:mobcontact/databases/db_helper.dart';
4  import 'package:mobcontact/models/contact_model.dart';
5  import 'package:mobcontact/pages/list_contact.dart';
6  import 'package:image_picker/image_picker.dart';
7  import 'package:path_provider/path_provider.dart';
8
9  class FormContact extends StatefulWidget {
10     final Contact contact;
11     FormContact(this.contact, {Key? key}) : super(key: key);
12     @override
13     State<FormContact> createState() => _FormContactState();
14 }
15
16 class _FormContactState extends State<FormContact> {
17     final DbHelper dbHelper = DbHelper();
18     File? imageFile;
19     final txtNama = TextEditingController();
20     final txtTelp = TextEditingController();
21     final txtEmail = TextEditingController();
22     final txtCompany = TextEditingController();
23     final txtPhoto = TextEditingController();
24     @override
25     void initState() {
26         super.initState();
27         if (widget.contact != null) {
28             txtNama.text = widget.contact.name ?? "";
29             txtTelp.text = widget.contact.mobileNo ?? "";
30             txtEmail.text = widget.contact.email ?? "";
```

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:mobcontact/databases/db_helper.dart';
import 'package:mobcontact/models/contact_model.dart';
import 'package:mobcontact/pages/list_contact.dart';
import 'package:image_picker/image_picker.dart';
import 'package:path_provider/path_provider.dart';

class FormContact extends StatefulWidget {
    final Contact contact;
    FormContact(this.contact, {Key? key}) : super(key: key);
    @override
    State<FormContact> createState() => _FormContactState();
}

class _FormContactState extends State<FormContact> {
    final DbHelper dbHelper = DbHelper();
    File? imageFile;
    final txtNama = TextEditingController();
```

```

final txtTelp = TextEditingController();
final txtEmail = TextEditingController();
final txtCompany = TextEditingController();
final txtPhoto = TextEditingController();
@override
void initState() {
  super.initState();
  if (widget.contact != null) {
    txtNama.text = widget.contact.name ?? "";
    txtTelp.text = widget.contact.mobileNo ?? "";
    txtEmail.text = widget.contact.email ?? "";
    txtCompany.text = widget.contact.company ?? "";
  }
}

Future<String> appPath() async {
  Directory appDocDir = await getApplicationDocumentsDirectory();
  return appDocDir.path;
}

getFromGallery() async {
  final XFile? image =
    await ImagePicker().pickImage(source: ImageSource.gallery);
  if (image == null) return;
  appPath().then((value) {
    String appPath = value;
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      // ignore: unnecessary_null_comparison
      title: widget.contact.id != null
        ? const Text("Update Contact")
        : const Text("Add Contact")),
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.symmetric(vertical: 8, horizontal: 16),
        child: Column(
          children: [
            TextField(
              keyboardType: TextInputType.name,
              controller: txtNama,

```

```

        decoration: const InputDecoration(
          labelText: "Nama",
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.person)),
      ),
      const SizedBox(height: 12),
      TextField(
        keyboardType: TextInputType.phone,
        controller: txtTelp,
        decoration: const InputDecoration(
          labelText: "Nomor HP",
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.phone)),
      ),
      const SizedBox(height: 12),
      TextField(
        keyboardType: TextInputType.emailAddress,
        controller: txtEmail,
        decoration: const InputDecoration(
          labelText: "Email",
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.email)),
      ),
      const SizedBox(height: 12),
      TextField(
        keyboardType: TextInputType.streetAddress,
        controller: txtCompany,
        decoration: const InputDecoration(
          labelText: "Nama Perusahaan",
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.home)),
      ),
      const SizedBox(height: 12),
      TextField(
        controller: txtPhoto,
        decoration: InputDecoration(
          labelText: "Photo",
          border: const OutlineInputBorder(),
          prefixIcon: const Icon(Icons.photo),
          suffixIcon: IconButton(
            icon: const Icon(Icons.photo_camera),
            onPressed: () {
              getFromGallery();
              print("pick image");
            },
          ),
        ),
      ),

```

```

    )),
    const SizedBox(height: 16),
    ElevatedButton(
      onPressed: () async {
        Contact data = Contact(
          name: txtNama.text,
          mobileNo: txtTelp.text,
          email: txtEmail.text,
          company: txtCompany.text,
          photo: txtPhoto.text);
        if (widget.contact.id != null) {
          data.id = widget.contact.id;
          await dbHelper.update(data);
        } else {
          await dbHelper.save(data);
        }
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => const ListContact()));
      },
      child: widget.contact.id != null
        ? const Text("Update Contact")
        : const Text("Save Contact"))
  ],
),
),
);
}
}

```

Sekian dari. Wassalamualaikum Warahmatullahi Wabarakatuh