GUIDE. A04

Capaian

1. Mahasiswa mampu membuat fungsi fungsi CRUD dengan dilengkapi error handling
2. Mahasiswa mampu membuat komponen UI yang biasa digunakan untuk melakukan operasi CRUD database

Dokumen ini bertujuan untuk menjadi dokumen pengantar mataeri pembelajaran Pemrograman Flutter dengan Database SQlite. Mahasiswa diharapkan mampu membuat widget komponen untuk *retrieve* data dari database menggunakan ListView builder.

Membuat Class ContactList pada File contact_list.dart

1. Import Class DbHelper, Contact, package material, dan Class ContactForm.
2. Buat class ContactList pada file contact_list.dart yang merupakan inheritance dari StatefullWidget dan memiliki konstruktor berparameter key.

```
const ContactList({Key? key}) : super(key: key);
```

3. Tambahkan anotasi '@override', dan buat metode createState yang mengembalikan instance kelas ContactListState.

Membuat Class ContactListState pada File contact_list.dart

1. Buat class ContactListState pada file yang sama. Class ini merupakan turunan dari State ContactList.
2. Inisialisai dua variabel berikut

Tipe data, Kata kunci	Nama Variabel	Value
DbHelper	db	DbHelper()
List<Contact>	listContact	[]

3. Tambahkan anotasi @override setelah deklarasi dan inisialisasi variabel. Lalu pada method void initState() tambahkan super.initState()
4. Buat metode 'build' yang mewarisi dari metode 'build' superclass menggunakan anotasi @override.
5. Di dalam metode 'build', kode mengembalikan widget 'Scaffold'. Scaffold merupakan widget yang umum digunakan untuk struktur dasar membuat aplikasi desain material.
6. Tambahkan beberapa parameter berikut ini:

Parameter	Widget	Parameter Widget	Value
key	Key	-	('scaffold_contactlist')
appBar	AppBar()	title	Center()

body	ListView.builder()	itemCount itemBuilder	lisContact.length (context,i){}
floatingActionButton	FloatingActionButton()	key child onPressed	Key('button_add') Icon(Icons.add,key: Key('icon_add')) _openFormCreate

7. Pada widget Center tambahkan beberapa parameter berikut ini:

Parameter	Widget	Parameter Widget	Value
key	Key()	-	Key(‘center_text_appbar’)
child	Text()	- key	ContactList Key(‘title_appbar_contactlist’)

8. Pada parameter widget itemBuilder di ListView.builder deklarasikan sebuah variabel dengan rincian sebagai berikut:
    a. Tipe data : Contact
    b. Nama variabel : c
    c. Nilai atau value: listContact[I]
9. Kemudian kembalikan fungsi atau method ListView.builder() dengan widget Padding dengan rincian informasi seperti dibawah ini:
    a. key: Key(‘padding_contactlist’)
    b. padding: EdgeInsets.only(top: 20)
    c. child: ListTile()

```
 return Padding(
              key: const …,
                padding: const EdgeInsets.only(
                    top: 20
                ),
                child: ListTile(
                  key:Key('litstile_$i')
));
```

10. Pada widget ListTile tambahkan parameter berikut ini berserta valuenya

Parameter	Widget	Parameter	Value
leading	Icon()	- key size	Icons.person Key(‘icon_person’) 50
title	Text()	-	'${c.name}'

		key	Key(‘text_name’)
subtitle	Column()		
trailing	FittedBox()		

11. Pada widget Column yang ada di ListTile memiliki parameter berikut ini

Parameter	Widget	Value
key	-	Key(‘column_contactlist’)
mainAxisAlignment	-	MainAxisAlignment.start
crossAxisAlignment	-	CrossAxisAlignment.start
children	Padding() Padding() Padding()	

12. Untuk setiap widget Padding yang ada pada parameter children widget Column, memiliki detail seperti dibawah ini.
    a. key: Key('padding_email')
    b. padding: const EdgeInsets.only(top:8)
    c. child: Text("Email: ${c.email}", key: Key('text_email_list'),)
13. Ulangi langkah 12 untuk membuat 2 padding lainnya.Sesuaikan pada parameter key, dan yang ada pada widget Text untuk data number dan data company.
14. Berikut contoh dari kodenya:

```
 Padding(
 key: const Key('...'), //disesuaikan padding_number,
padding_company
 padding: const EdgeInsets.only(top: 8),
 child: Text("Email: ${c.email}", //disesuaikan
 key: Key('text_email_list'),),), //disesuaikan
```

15. Pada widget FittedBox yang ada di ListTile memiliki parameter berikut ini

Parameter	Widget	Value
fit	-	Boxfit.fill
key	-	Key('fittedbox_contactlist')
child	Row()	

16. Pada parameter child widget Row memiliki parameter berikut:

Parameter	Widget	Value
key	-	Key(‘row_contactlist’)
children	IconButton() IconButton()	MainAxisAlignment.start

17. IconButton pertama memiliki parameter berikut:

Parameter	Widget	Value
key	-	Key('iconbutton_edit_$i')
onPressed	-	_openFormEdit(c)
icon	Icon()	Icons.edit, key: Key('icon_edit')

18. Berikut template kode yang dapat digunakan untuk langkah 17

```
IconButton(
  key: …,
  onPressed: () async {
       ..;
   },
   icon: const Icon(…, key: …)),
```

19. Pada IconButton kedua memiliki parameter berikut ini

Parameter	Widget	Value
key	-	Key('textdeletebutton')
onPressed	-	(){}
icon	Icon()	Icons.delete, key: Key('icon_delete')

20. Lalu pada parameter onPressed deklarasikan variabel delete yang memiliki tipe data AlertDialog dan value AlertDialog. Kode ini sebagai pesan konfirmasi jika user ingin menghapus data.

```
AlertDialog delete = AlertDialog();
```

21. Pada AlertDialog memiliki parameter berikut ini:

Parameter	Widget	Paremter	Value
key	-	-	Key('alertdialog')
title	Text()	-	“Information”

content	SizedBox()	key height child	Key('sizedbox') 100 Column()
actions	TextButton() TextButton()	-	-

22. Pada widget Column tambahkan detail seperti ini

```
children: const [
  Text(
  "Are you sure to delete this contact?",
   key: Key('confirmation_text'),)
```

23. Widget TextButtonPertama tambahkan informasi berikut:
    a. key: Key('textdeletebutton')
    b. onPressed: () async {_deleteContact(c,I); Navigator.pop(context);},
    c. child: Text("Yes")
24. Widget TextButtonPertama tambahkan informasi berikut:
    a. key: Key('textcancelbutton')
    b. onPressed: () {Navigator.pop(context);},
    c. child: Text("No")
25. Tambahkan kode ini dibawah widget AlertDialog .

```
showDialog(context: context,builder: (context) => delete);
```

26. Buat fungsi _getAllContact, _openFormCreate(), _openFormEdit(), _deleteContact() diluar fungsi '`Widget build(BuilContext context){}`'
    a. Fungsi _getAllContact memiliki tipe data Future<void> dan bersifat asynchronous. Fungsi ini digunakan untuk mengambil data dari database. Tambahkan fungsi _getAllContact pada fungsi initState()

```
Future<void> _getAllContact() async {
  var list = await db.getAllContact();
  setState(() {
    listContact.clear();
    for (var data in list!) {
      listContact.add(Contact.fromMap(data));
    }
  });
}
```

b. Fungsi _openFormCreate() bertipe Future<void> dan bersifat asnchronous. Fungsi ini digunakan untuk beralih ke halaman ContactForm

```
Future<void> _openFormCreate() async {
  var result = await Navigator.push(
      context, MaterialPageRoute(builder: (context) => ContactForm()));
  if (result == 'save') {
    await _getAllContact();
  }
}
```

c. Fungsi _openFormEdit bertipe data Future<void>, memiliki parameter dari Contact, serta bersifat asynchronous. Fungsi ini digunakan untuk membuka halam form dengan membawa data contact yang akan diubah.

```
Future<void> _openFormEdit(Contact contact) async {
  var result = await Navigator.push(context,
      MaterialPageRoute(builder: (context) => ContactForm(contact: contact)));
  if (result == 'update') {
    await _getAllContact();
  }
}
```

d. Fungsi _deleteContact() menerima dua parameter yaitu parameter dari class Contact dan parameter berupa integer. Fungsi ini digunakan untuk menghapus data contact.

```
Future<void> _deleteContact(Contact contact, int position) async {
  await db.deleteContact(contact.id!);
  setState(() {
    listContact.removeAt(position);
  });
}
```

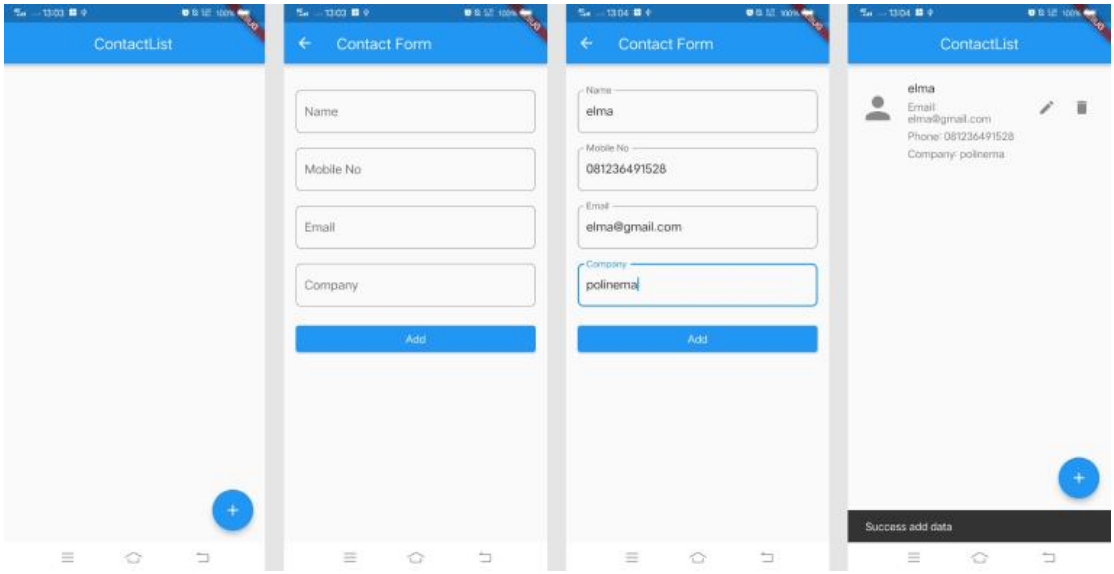
Mengubah Parameter Home pada class MyApp di file main.dart

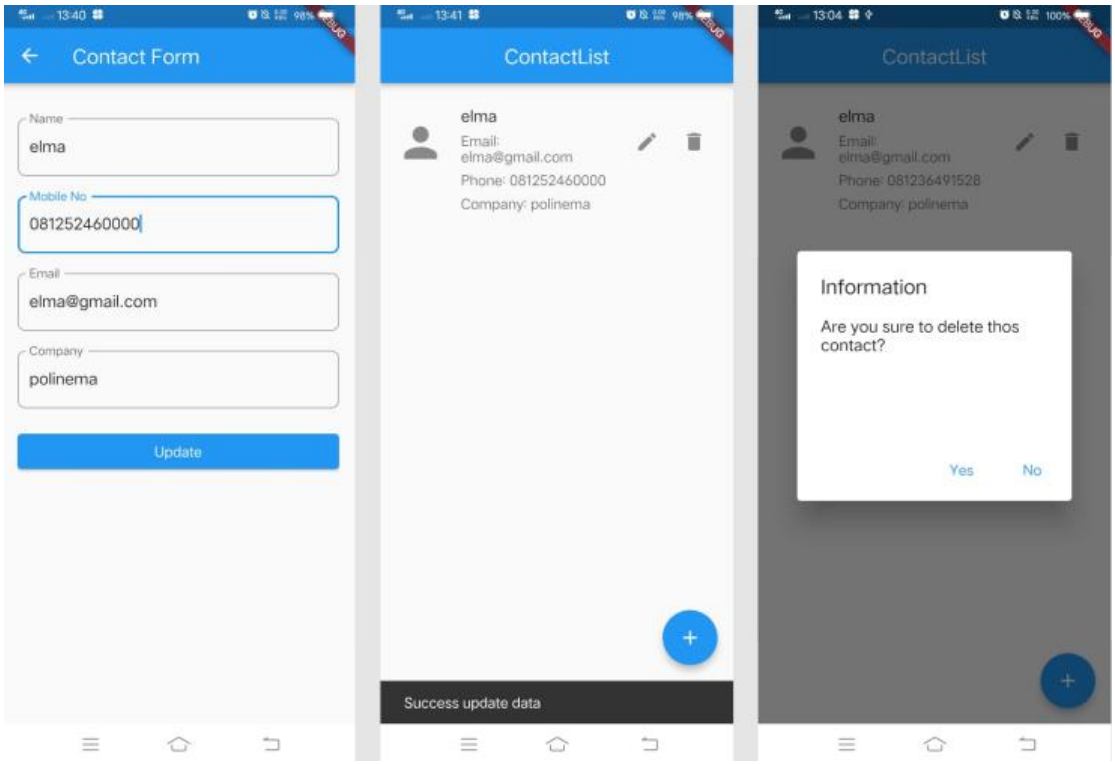
1. Tambahkan parameter key pada MaterialApp di kelas MyApp yang memiliki value Key(‘material_app)

2. Pada parameter Home ubah bilai menjadi seperti berikut ini:

```
home: ContactList(key: Key('contact_list'),)
```

Output Aplikasi Simple Database





Testing Project

1. Unduh code file guide4_1_test.dart dan guide4_2_test.dart, kemudian pindahkan ke folder test pada project anda
2. Jalankan test file guide4_1_test.dart terlebih dahulu. Jika ada yang gagal maka tama
3. Pada class ContactList, tambahkan code berikut sebagai value dari variabel listContact

```
Contact(name: "Risa", email: "risa@gmail.com", number:
"087654389090", company: "Polinema")
```

4. Lalu jalankan test guide4_2_test.dart. Jika berhasil semua maka akan muncul keterangan ‘Test Success’ dan jika ada yang gagal maka akan muncul keterangan ‘Test Fail’ disertai keterangan bagian mana yang fail.