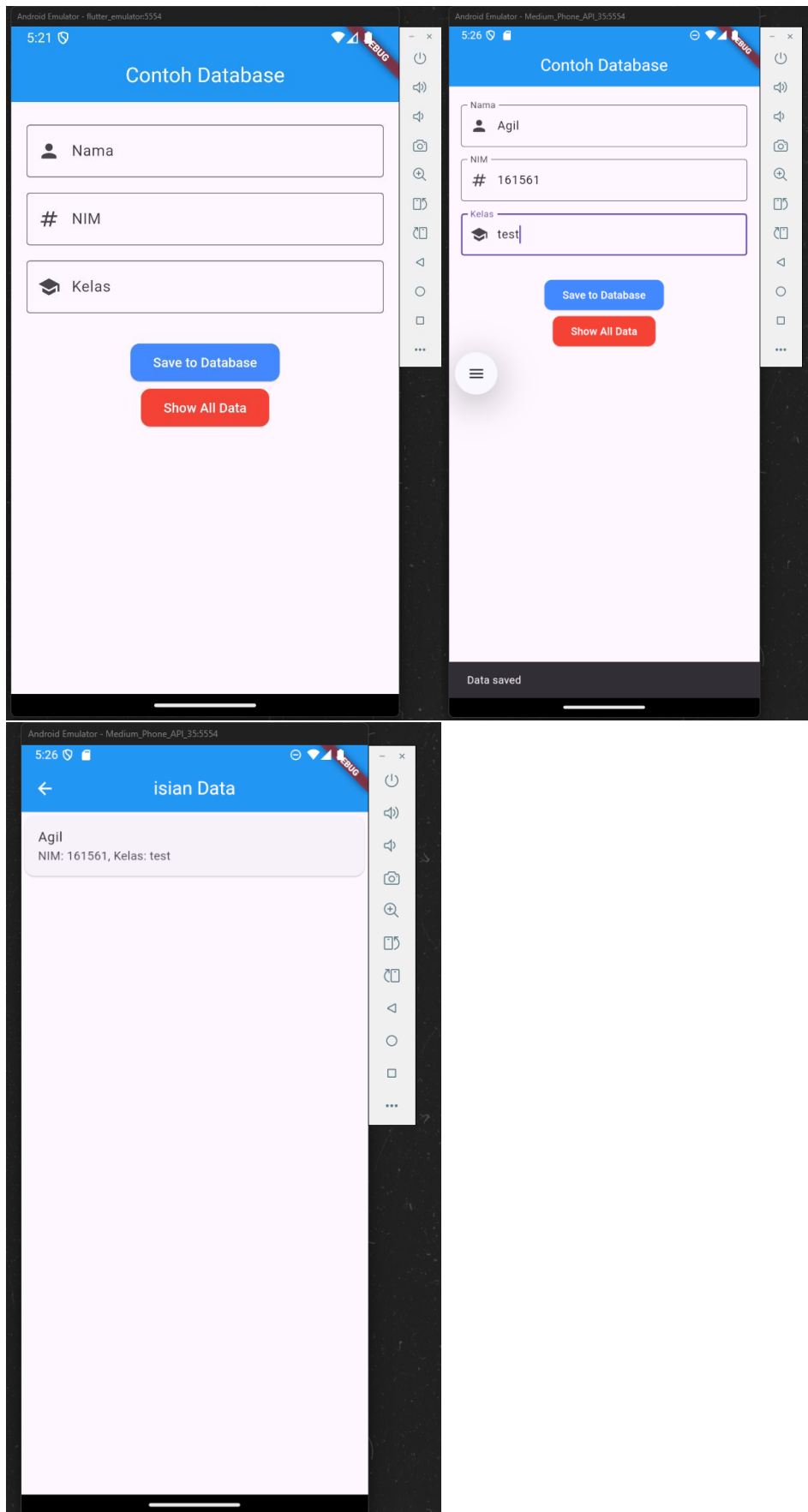


Nama : Fadhila Agil Permana

NIM : 2211104006

Kelas : SE06-1

## OUTPUT



# Source Code

## Main.dart

```
1 // ignore_for_file: flutter_style_lints
2
3 import 'package:flutter/material.dart';
4 import 'package:supabase_flutter/supabase_flutter.dart';
5 import 'package:supabase_flutter/supabase_flutter.dart';
6
7 void main() {
8   runApp(const MyApp());
9 }
10
11 class MyApp extends StatelessWidget {
12   const MyApp({super.key});
13
14   @override
15   Widget build(BuildContext context) {
16     return MaterialApp(
17       title: 'Conch Database',
18       theme: ThemeData(
19         primarySwatch: Colors.blue,
20       ),
21       home: MyHomePage(),
22     );
23   }
24 }
25
26 class MyHomePage extends StatelessWidget {
27   const MyHomePage({super.key});
28
29   @override
30   Widget build(BuildContext context) {
31     return Scaffold(
32       appBar: AppBar(
33         title: 'Conch Database',
34       ),
35       body: MyHomePageState(),
36     );
37   }
38 }
39
40 class MyHomePageState extends State<MyHomePage> {
41   final _formKey = GlobalKey<FormState>();
42   final _nameController = TextEditingController();
43   final _emailController = TextEditingController();
44   final _passwordController = TextEditingController();
45   final _confirmPasswordController = TextEditingController();
46   final _isLoading = false;
47
48   @override
49   void initState() {
50     super.initState();
51     _isLoading = false;
52   }
53
54   @override
55   void dispose() {
56     _nameController.dispose();
57     _emailController.dispose();
58     _passwordController.dispose();
59     _confirmPasswordController.dispose();
60   }
61
62   Future<void> _addConch() async {
63     try {
64       await supabase.from('conchs').insert([
65         {
66           'name': _nameController.text,
67           'email': _emailController.text,
68           'password': _passwordController.text,
69           'confirm_password': _confirmPasswordController.text,
70         },
71       ]);
72     } catch (e) {
73       // Handle error
74     }
75   }
76
77   @override
78   Widget build(BuildContext context) {
79     return Scaffold(
80       appBar: AppBar(
81         title: 'Conch Database',
82       ),
83       body: Padding(
84         padding: const EdgeInsets.all(16.0),
85         child: Form(
86           key: _formKey,
87           child: Column(
88             children: [
89               TextFormField(
90                 textInputAction: TextInputAction.next,
91                 controller: _nameController,
92                 decoration: InputDecoration(
93                   labelText: 'Name',
94                 ),
95                 validator: (value) {
96                   if (value == null || value.isEmpty) {
97                     return 'Please enter name';
98                   }
99                 },
100               ),
101               TextFormField(
102                 textInputAction: TextInputAction.next,
103                 controller: _emailController,
104                 decoration: InputDecoration(
105                   labelText: 'Email',
106                 ),
107                 validator: (value) {
108                   if (value == null || value.isEmpty) {
109                     return 'Please enter email';
110                   }
111                 },
112               ),
113               TextFormField(
114                 textInputAction: TextInputAction.next,
115                 controller: _passwordController,
116                 decoration: InputDecoration(
117                   labelText: 'Password',
118                 ),
119                 validator: (value) {
120                   if (value == null || value.isEmpty) {
121                     return 'Please enter password';
122                   }
123                 },
124               ),
125               TextFormField(
126                 textInputAction: TextInputAction.done,
127                 controller: _confirmPasswordController,
128                 decoration: InputDecoration(
129                   labelText: 'Confirm Password',
130                 ),
131                 validator: (value) {
132                   if (value == null || value.isEmpty) {
133                     return 'Please enter confirm password';
134                   }
135                 },
136               ),
137             ],
138           ),
139         ),
140       ),
141       floatingActionButton: FloatingActionButton(
142         onPressed: _addConch,
143         tooltip: 'Add Conch',
144       ),
145     );
146   }
147 }
148
149 class MyHomePageState extends State<MyHomePage> {
150   final _formKey = GlobalKey<FormState>();
151   final _nameController = TextEditingController();
152   final _emailController = TextEditingController();
153   final _passwordController = TextEditingController();
154   final _confirmPasswordController = TextEditingController();
155   final _isLoading = false;
156
157   @override
158   void initState() {
159     super.initState();
160     _isLoading = false;
161   }
162
163   @override
164   void dispose() {
165     _nameController.dispose();
166     _emailController.dispose();
167     _passwordController.dispose();
168     _confirmPasswordController.dispose();
169   }
170
171   Future<void> _addConch() async {
172     try {
173       await supabase.from('conchs').insert([
174         {
175           'name': _nameController.text,
176           'email': _emailController.text,
177           'password': _passwordController.text,
178           'confirm_password': _confirmPasswordController.text,
179         },
180       ]);
181     } catch (e) {
182       // Handle error
183     }
184   }
185
186   @override
187   Widget build(BuildContext context) {
188     return Scaffold(
189       appBar: AppBar(
190         title: 'Conch Database',
191       ),
192       body: Padding(
193         padding: const EdgeInsets.all(16.0),
194         child: Form(
195           key: _formKey,
196           child: Column(
197             children: [
198               TextFormField(
199                 textInputAction: TextInputAction.next,
200                 controller: _nameController,
201                 decoration: InputDecoration(
202                   labelText: 'Name',
203                 ),
204                 validator: (value) {
205                   if (value == null || value.isEmpty) {
206                     return 'Please enter name';
207                   }
208                 },
209               ),
210               TextFormField(
211                 textInputAction: TextInputAction.next,
212                 controller: _emailController,
213                 decoration: InputDecoration(
214                   labelText: 'Email',
215                 ),
216                 validator: (value) {
217                   if (value == null || value.isEmpty) {
218                     return 'Please enter email';
219                   }
220                 },
221               ),
222               TextFormField(
223                 textInputAction: TextInputAction.next,
224                 controller: _passwordController,
225                 decoration: InputDecoration(
226                   labelText: 'Password',
227                 ),
228                 validator: (value) {
229                   if (value == null || value.isEmpty) {
230                     return 'Please enter password';
231                   }
232                 },
233               ),
234               TextFormField(
235                 textInputAction: TextInputAction.done,
236                 controller: _confirmPasswordController,
237                 decoration: InputDecoration(
238                   labelText: 'Confirm Password',
239                 ),
240                 validator: (value) {
241                   if (value == null || value.isEmpty) {
242                     return 'Please enter confirm password';
243                   }
244                 },
245               ),
246             ],
247           ),
248         ),
249       ),
250       floatingActionButton: FloatingActionButton(
251         onPressed: _addConch,
252         tooltip: 'Add Conch',
253       ),
254     );
255   }
256 }
```

## db\_helper.dart

```
1 import 'package:sqflite/sqflite.dart';
2 import 'package:path/path.dart';
3
4 class DatabaseHelper {
5   static final DatabaseHelper _instance = DatabaseHelper._internal
6   () => DatabaseHelper._instance;
7   static Database? _database;
8
9   DatabaseHelper._internal();
10
11   Future<Database> get database async {
12     if (_database != null) return _database!;
13     _database = await _initDatabase();
14     return _database!;
15   }
16
17   Future<Database> _initDatabase() async {
18     String path = join(await getDatabasesPath(), 'my_database.db');
19     return await openDatabase(
20       path,
21       version: 1,
22       onCreate: _onCreate,
23     );
24   }
25
26   Future<void> _onCreate(Database db, int version) async {
27     await db.execute("""
28       CREATE TABLE my_table (
29         id INTEGER PRIMARY KEY AUTOINCREMENT,
30         nama TEXT,
31         nim TEXT,
32         kelas TEXT
33       )
34     """);
35   }
36
37   Future<void> insert(String nama, String nim, String kelas) async {
38     final db = await database;
39     await db.insert(
40       'my_table',
41       {'nama': nama, 'nim': nim, 'kelas': kelas},
42       conflictAlgorithm: ConflictAlgorithm.replace,
43     );
44   }
45
46   Future<List<Map<String, dynamic>>> queryAllRows() async {
47     final db = await database;
48     return await db.query('my_table');
49   }
50
51   Future<void> deleteDatabaseFile() async {
52     String path = join(await getDatabasesPath(), 'my_database.db');
53     if (await databaseExists(path)) {
54       await deleteDatabase(path);
55     }
56   }
57 }
58
```

## Showdata.dart

```
1  import 'package:flutter/material.dart';
2  import 'package:yupi/helper/db_helper.dart';
3
4  class SecondScreen extends StatelessWidget {
5    final DatabaseHelper _dbHelper = DatabaseHelper();
6
7    SecondScreen({super.key});
8
9    @override
10   Widget build(BuildContext context) {
11     return Scaffold(
12       appBar: AppBar(
13         centerTitle: true,
14         backgroundColor: Colors.blue,
15         foregroundColor: Colors.white,
16         title: const Text('isian Data'),
17       ),
18       body: FutureBuilder<List<Map<String, dynamic>>>>(
19         future: _dbHelper.queryAllRows(),
20         builder: (context, snapshot) {
21           if (snapshot.connectionState == ConnectionState.waiting)
22           {
23             return const Center(child: CircularProgressIndicator());
24           } else if (snapshot.hasError) {
25             return Center(child: Text('Error: ${snapshot.error}'));
26           } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
27             return const Center(child: Text('No data found'));
28           } else {
29             return ListView.builder(
30               itemCount: snapshot.data!.length,
31               itemBuilder: (context, index) {
32                 final row = snapshot.data![index];
33                 return Card(
34                   child: ListTile(
35                     title: Text(row['nama']),
36                     subtitle:
37                       Text('NIM: ${row['nim']}, Kelas: ${row['kelas']}'),
38                   ),
39                 );
40               },
41             );
42           }
43         ),
44       );
45     }
46   }
47 }
```