

Dokumentasi Crud ATM data diri

Nama : Muhamad Kurniawan

Nim : 11203362310108

Database

```
CREATE DATABASE data_diri_db;

USE data_diri_db;

CREATE TABLE tb_data_diri (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nama VARCHAR(100) NOT NULL,
  umur INT NOT NULL,
  hobi VARCHAR(100),
  asal VARCHAR(100),
  tinggi_badan DECIMAL(5,2), -- Contoh: 170.50 cm
  berat_badan DECIMAL(5,2)  -- Contoh: 65.75 kg
);
```

Backend Crud

db.php

```
<?php
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Authorization, X-Request-With");

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "data_diri_db"; // ganti sesuai nama database yang baru
```

```

$conn = new mysqli($servername, $username, $password, $dbname);

// Cek koneksi
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

```

create.php

```

<?php
include 'db.php';

// Ambil data dari form POST
$nama = $_POST['nama'];
$umur = $_POST['umur'];
$hobi = $_POST['hobi'];
$asal = $_POST['asal'];
$tinggi_badan = $_POST['tinggi_badan'];
$berat_badan = $_POST['berat_badan'];

// Query insert ke tabel tb_data_dir
$sql = "INSERT INTO tb_data_diri (nama, umur, hobi, asal, tinggi_badan, berat_ba
VALUES ('$nama', $umur, '$hobi', '$asal', $tinggi_badan, $berat_badan)";

if ($conn->query($sql) === TRUE) {
    echo "Data diri berhasil ditambahkan";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>

```

read.php

```
<?php
include 'db.php';

$sql = "SELECT * FROM tb_data_diri";
$result = $conn->query($sql);

$data = array();
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $data[] = $row;
    }
}
echo json_encode($data);

$conn->close();
?>
```

update.php

```
<?php
include 'db.php';

$id = $_POST['id'];
$name = $_POST['nama'];
$umur = $_POST['umur'];
$hobi = $_POST['hobi'];
$asal = $_POST['asal'];
$tinggi_badan = $_POST['tinggi_badan'];
$berat_badan = $_POST['berat_badan'];

$sql = "UPDATE tb_data_diri
        SET nama='$name', umur=$umur, hobi='$hobi', asal='$asal', tinggi_badan=
        WHERE id=$id";
```

```

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>

```

delete.php

```

<?php
include 'db.php';

$id = $_POST['id'];

$sql = "DELETE FROM tb_data_diri WHERE id=$id";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>

```

persiapan untuk flutter

1. project baru flutter
2. menginstall http di vscode (di file pubget.yaml) dan menjalankan flutter pub get di terminal vscode
3. update code di main.dart

```

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter CRUD Data Diri',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() ⇒ _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  List dataDiri = [];
  String apiUrl = 'http://127.0.0.1/api-db-kampus/'; // Ganti IP sesuai kebutuhan

  @override
  void initState() {
    super.initState();
    fetchDataDiri();
  }
}

```

```

fetchDataDiri() async {
    final response = await http.get(Uri.parse(apiUrl + 'read.php'));
    if (response.statusCode == 200) {
        setState(() {
            dataDiri = json.decode(response.body);
        });
    } else {
        throw Exception('Gagal mengambil data');
    }
}

addDataDiri(Map data) async {
    data['id'] ??= "";
    final response = await http.post(Uri.parse(apiUrl + 'create.php'), body: data);
    if (response.statusCode == 200) {
        fetchDataDiri();
    } else {
        throw Exception('Gagal menambahkan data');
    }
}

updateDataDiri(Map data) async {
    final response = await http.post(Uri.parse(apiUrl + 'update.php'), body: data);
    if (response.statusCode == 200) {
        fetchDataDiri();
    } else {
        throw Exception('Gagal memperbarui data');
    }
}

deleteDataDiri(String id) async {
    final response = await http.post(Uri.parse(apiUrl + 'delete.php'), body: {'id': id});
    if (response.statusCode == 200) {
        fetchDataDiri();
    } else {

```

```

        throw Exception('Gagal menghapus data');
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Data Diri'),
        ),
        body: ListView.builder(
            itemCount: dataDiri.length,
            itemBuilder: (context, index) {
                final item = dataDiri[index];
                return Card(
                    child: ListTile(
                        title: Text(item['nama']),
                        subtitle: Text("Asal: ${item['asal']}, Hobi: ${item['hobi']}"),
                        trailing: Row(
                            mainAxisAlignment: MainAxisAlignment.min,
                            children: [
                                IconButton(
                                    icon: Icon(Icons.edit, color: Colors.orange),
                                    onPressed: () {
                                        showDialog(
                                            context: context,
                                            builder: (context) {
                                                return DataDiriForm(
                                                    dataDiri: item,
                                                    onSubmit: updateDataDiri,
                                                );
                                            },
                                        );
                                    },
                                ),
                                IconButton(

```

```

        icon: Icon(Icons.delete, color: Colors.red),
        onPressed: () {
          deleteDataDiri(item['id']);
        },
      ),
    ],
  ),
);
},
),
floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.green,
  onPressed: () {
    showDialog(
      context: context,
      builder: (context) {
        return DataDiriForm(onSubmit: addDataDiri);
      },
    );
  },
  child: Icon(Icons.add),
),
);
}
}

```

```

class DataDiriForm extends StatefulWidget {
  final Map? dataDiri;
  final Function(Map) onSubmit;

  DataDiriForm({this.dataDiri, required this.onSubmit});

  @override
  _DataDiriFormState createState() ⇒ _DataDiriFormState();
}

```



```

class _DataDiriFormState extends State<DataDiriForm> {
  final _formKey = GlobalKey<FormState>();
  TextEditingController _namaController = TextEditingController();
  TextEditingController _umurController = TextEditingController();
  TextEditingController _hobiController = TextEditingController();
  TextEditingController _asalController = TextEditingController();
  TextEditingController _tinggiController = TextEditingController();
  TextEditingController _beratController = TextEditingController();

  @override
  void initState() {
    super.initState();
    if (widget.dataDiri != null) {
      _namaController.text = widget.dataDiri!['nama'];
      _umurController.text = widget.dataDiri!['umur'].toString();
      _hobiController.text = widget.dataDiri!['hobi'];
      _asalController.text = widget.dataDiri!['asal'];
      _tinggiController.text = widget.dataDiri!['tinggi_badan'].toString();
      _beratController.text = widget.dataDiri!['berat_badan'].toString();
    }
  }

  @override
  Widget build(BuildContext context) {
    return AlertDialog(
      title: Text(widget.dataDiri == null ? 'Tambah Data Diri' : 'Edit Data Diri'),
      content: SingleChildScrollView(
        child: Form(
          key: _formKey,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              _buildTextField(_namaController, 'Nama'),
              _buildTextField(_umurController, 'Umur', keyboardType: TextInputType.n
              _buildTextField(_hobiController, 'Hobi'),

```

```

        _buildTextField(_asalController, 'Asal'),
        _buildTextField(_tinggiController, 'Tinggi Badan (cm)', keyboardType: Te,
        _buildTextField(_beratController, 'Berat Badan (kg)', keyboardType: TextI
    ],
  ),
),
),
actions: [
  TextButton(
    child: Text('Batal'),
    onPressed: () ⇒ Navigator.of(context).pop(),
  ),
  TextButton(
    child: Text('Simpan'),
    onPressed: () {
      if (_formKey.currentState!.validate()) {
        final data = {
          'id': widget.dataDiri?['id'],
          'nama': _namaController.text,
          'umur': _umurController.text,
          'hobi': _hobiController.text,
          'asal': _asalController.text,
          'tinggi_badan': _tinggiController.text,
          'berat_badan': _beratController.text,
        };
        widget.onSubmit(data);
        Navigator.of(context).pop();
      }
    },
  ),
],
);
}

```

```

Widget _buildTextField(TextEditingController controller, String label,
  {TextInputType keyboardType = TextInputType.text}) {

```

```

return Padding(
  padding: const EdgeInsets.symmetric(vertical: 4),
  child: TextFormField(
    controller: controller,
    keyboardType: keyboardType,
    decoration: InputDecoration(labelText: label),
    validator: (value) => (value == null || value.isEmpty) ? '$label tidak boleh kos
  ),
);
}

@override
void dispose() {
  _namaController.dispose();
  _umurController.dispose();
  _hobiController.dispose();
  _asalController.dispose();
  _tinggiController.dispose();
  _beratController.dispose();
  super.dispose();
}
}

```



