MODUL 8
NETWORKING PADA FLUTTER

A. Capaian Pembelajaran

Mahasiswa dapat memahami tentang peran penting jaringan dalam aplikasi menggunakan Flutter dan mampu membangun *server backend* serta menjalankan *server* lokal dengan menggunakan *server* JSON untuk membangun REST API dengan operasi CRUD.

B. Tujuan Pembelajaran

Pada modul 8, akan membahas secara detail hal-hal yang diperlukan dalam rangka membuat aplikasi *smartphone* berbasis Android dengan *framework* Flutter, dengan tujuan:

1. Dapat membuat API.
2. Dapat mengimplementasi API pada Flutter.

C. Teori & Praktek

API adalah singkatan dari *Application Programming Interface* yaitu sebuah software yang memungkinkan para *developer* untuk mengintegrasikan dan mengizinkan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain. Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda tersebut, Tujuan penggunaan API lainnya yaitu untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah *function* yang terpisah sehingga para *developer* tidak perlu lagi membuat fitur yang serupa.

JSON *(JavaScript Object Notation)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON sebagai format untuk bertukar data *client* dan *server* atau antar aplikasi. Contoh: RESTful API.

Berikut ini beberapa penerapan JSON:

1. JSON sebagai tempat menyimpan data, contoh: *Database* Mongodb;
2. JSON digunakan untuk menyimpan konfigurasi *project*, contoh: *file composer.json* pada *project* PHP dan *package.json* pada Nodejs;
3. JSON digunakan untuk menyimpan konfigurasi dan penyimpanan data pada Hugo;

4. JSON digunakan untuk menyimpan konfigurasi *project* pada Nodejs;
5. JSON digunakan untuk menyimpan data *manifest*.

Struktur JSON dapat dilihat pada gambar berikut:

Gambar 8.1 Struktur JSON

Kerjakan Langkah-langkah praktikum di bawah ini:

Buat *database* dengan nama **db_mahasiswa**. Kemudian buat tabel seperti di bawah ini.

```
CREATE TABLE IF NOT EXISTS `tb_mahasiswa` (
        `nim` VARCHAR(50) NOT NULL PRIMARY KEY,
        `nama` text NOT NULL,
         `kelas` VARCHAR(10),
        `jurusan` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

1. Buat *file* **config.php**

```php
<?php
$conn = mysqli_connect("localhost","root","");
mysqli_select_db($conn, "db_mahasiswa");
?>
```

2. Buat *file* untuk memasukan data dengan nama **insert.php**

```php
<?php
include_once('config.php');
if($_SERVER['REQUEST_METHOD'] == "POST"){
    // Get data from the REST client
    $nim = isset($_POST['nim']) ? mysqli_real_escape_string($conn, $_POST['
nim']) : "";
    $nama = isset($_POST['nama']) ? mysqli_real_escape_string($conn, $_POST
['nama']) : "";
    $kelas = isset($_POST['kelas']) ? mysqli_real_escape_string($conn, $_PO
ST['kelas']) : "";
    $jurusan = isset($_POST['kelas']) ? mysqli_real_escape_string($conn, $_
POST['jurusan']) : "";
```

```
    // Insert data into database
    $sql = "INSERT INTO tb_mahasiswa(nim,nama,kelas,jurusan) VALUES('$nim',
'$nama','$kelas','$jurusan')";
    $post_data_query = mysqli_query($conn, $sql);
    if($post_data_query){
        $json = array("status" => 1, "Success" => "Insert Success");
    }
    else{
        $json = array("status" => 0, "Error" => "Error Insert Data");
    }
}
else{
    $json = array("status" => 0, "Info" => "Request method not accepted!");
}
@mysqli_close($conn);
// Set Content-type to JSON
header('Content-type: application/json');
echo json_encode($json);
```

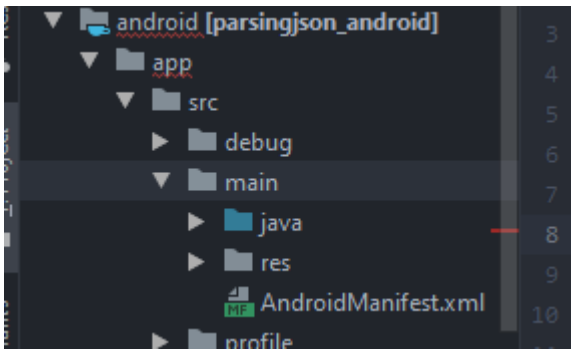
3. Buat *file* untuk menampilkan data dengan nama **list_mahasiswa.php**

```
<?php
    include_once('config.php');

    $sql = "SELECT * FROM tb_mahasiswa";
    $get_data_query = mysqli_query($conn, $sql) or die(mysqli_error($conn))
;
        if(mysqli_num_rows($get_data_query)!=0){
        $result = array();

        while($r = mysqli_fetch_array($get_data_query)){
            extract($r);
            $result[] = array("nim" => $nim, "nama" => $nama, 'kelas' => $k
elas,'jurusan'=> $jurusan);
        }
        $json = array("status" => 1, "info" => $result);
    }
    else{
        $json = array("status" => 0, "error" => "mahasiswa not found!");
    }
@mysqli_close($conn);
// Set Content-type to JSON
header('Content-type: application/json');
echo json_encode($json);
```

4. *File* tersebut disimpan dalam satu *folder* di *service*, dengan menggunakan XAMPP dan disimpan dalam folder **api-mahasiswa**.
5. Buka ***cmd*** lalu ketik **ipconfig**

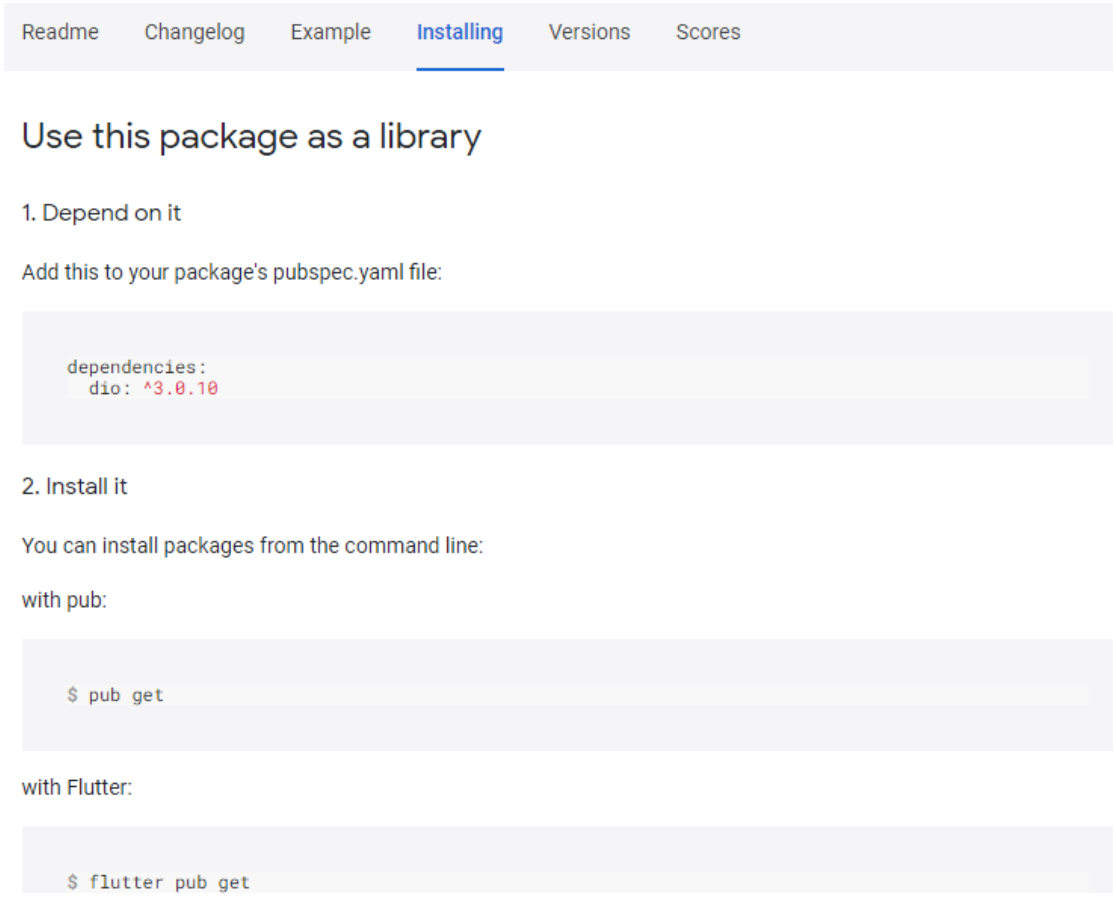
```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::445c:2a47:6f8b:8e8f%6
   IPv4 Address. . . . . . . . . . . : 192.168.43.27
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.43.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

6. Catat IP-nya untuk dijadikan domain kita.
7. Buat *project* Flutter dengan nama **parsing_json**. Pada **AndroidManifest.xml** tambahkan *permission* untuk penggunaan Internet. Lokasinya ada dibagian *android/app/src/main/androidManifest.xml*

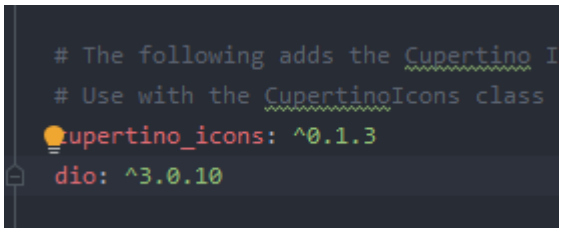



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="grosircenter.parsingjson">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application
that
         calls FlutterMain.startInitialization(this); in its onCreate
method.
         In most cases you can leave this as-is, but you if you want to
provide
         additional functionality it is fine to subclass or reimplement
         FlutterApplication and put your custom class here. -->
//tambahkan file tersebut
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:name="io.flutter.app.FlutterApplication"
```

8. Tambahkan juga *package library* ***dio*** untuk menjembatani API dan *mobile* aplikasi kita. Untuk *link* bisa di ***pub.dev***, kemudian cari dengan nama ***dio***.



Ikuti instruksinya, untuk penggunaannya tambahkan di *file* ***pubspec.yaml***



Ketik Flutter *pubget* pada *terminal*.

9. Pada *folder lib* buat satu *layout* dengan nama **v_json.dart**, kemudian buat *class statefullWidget*.

```
import 'package:flutter/material.dart';

class ViewJson extends StatefulWidget {
  @override
  _ViewJsonState createState() => _ViewJsonState();
}

class _ViewJsonState extends State<ViewJson> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Parsing Json"),),
    );
```

```
  }
}
```

10. Buat *variabel list* mahasiswa dan *dio* agar bisa digunakan untuk komunikasi dengan API.

```
class _ViewJsonState extends State<ViewJson> {

  Dio dio=new Dio();
  List<Mahasiswa> listMahasiswa=[];
```

11. Buat *class model* mahasiwa di luar *class statefullWidget*.

```
class Mahasiswa{
  final String nim;
  final String nama;
  final String kelas;
  final String jurusan;

  Mahasiswa({this.nim, this.nama, this.kelas, this.jurusan});
}
```

12. Buat fungsi ***get*** Mahasiwa di bawah *variable* yang telah dibuat.

```
Future<void> getMahasiswa()async{
  Response response;
  response=await dio.get("http://192.168.43.27/api-
mahasiswa/list_mahasiswa.php");
  if(response.data['status']==1){
    setState(() {
      for(int i=0;i<response.data['info'].length;i++){
        listMahasiswa.add(Mahasiswa(
          nim: response.data['info']['nim'],
          nama: response.data['info']['nama'],
          kelas: response.data['info']['kelas'],
          jurusan: response.data['info']['jurusan'],
        ));
      }
    });
  }
}
```

Method initState() panggil fungsi yang sudah dibuat tadi.

```
@override
void initState() {
  getMahasiswa();
  super.initState();
}
```

13. Buat *method* untuk *insert* data ke dalam API yang sudah ada.

```
Future<void> insertMahasiswa()async{
  Response response;
  response=await dio.post("http://192.168.43.27/api-
```

```
mahasiswa/list_mahasiswa.php",data: {
    "nim"   : "${_controllerNim.text}",
    "nama"   : "${_controllerNama.text}",
    "kelas"   : "${_controllerkelas.text}",
    "jurusan"   : "${_controllerjurusan.text}",
  });
  if(response.data['status']==1){
    _scaffold.currentState.showSnackBar(SnackBar(content: Text("Berhasil
disimpan")));
    setState(() {
      _controllerNim.text="";
      _controllerNama.text="";
      _controllerkelas.text="";
      _controllerjurusan.text="";
      getMahasiswa();
    });
  }
}
```

14. Tambahakan *variable* baru seperti di bawah ini.

```
TextEditingController _controllerNim=TextEditingController();
TextEditingController _controllerNama=TextEditingController();
TextEditingController _controllerkelas=TextEditingController();
TextEditingController _controllerjurusan=TextEditingController();
GlobalKey<ScaffoldState> _scaffold=new GlobalKey<ScaffoldState>();
```

15. Pasang *variable scaffold* pada *key scaffold* yang sudah dibuat.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _scaffold,
    appBar: AppBar(title: Text("Parsing Json"),),

  );
}
```

16. Buatlah *layout* seperti kode dibawah ini.

```
@override
  Widget build(BuildContext context) {
    return Scaffold(
      key: _scaffold,
      appBar: AppBar(
        title: Text("Parsing Json"),
      ),
      body: MediaQuery.removePadding(
        context: context,
        removeTop: true,
        child: ListView(
          padding: EdgeInsets.symmetric(horizontal: 15, vertical: 10),
          shrinkWrap: true,
          physics: ClampingScrollPhysics(),
          children: [
            Text("Form Input"),
            Divider(
              thickness: 1,
```

```
          ),
          SizedBox(
            height: 10,
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              Flexible(
                  child: TextField(
                controller: _controllerNim,
                decoration: InputDecoration(hintText: "Nim"),
              )),
              SizedBox(
                width: 10,
              ),
              Flexible(
                  child: TextField(
                controller: _controllerkelas,
                decoration: InputDecoration(hintText: "kelas"),
              )),
            ],
          ),
          SizedBox(
            height: 10,
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              Flexible(
                  child: TextField(
                controller: _controllerNama,
                decoration: InputDecoration(hintText: "Nama Mhs"),
              )),
              SizedBox(
                width: 10,
              ),
              Flexible(
                  child: TextField(
                controller: _controllerjurusan,
                decoration: InputDecoration(hintText: "Jurusan"),
              )),
            ],
          ),
          SizedBox(
            height: 10,
          ),
          RaisedButton(
            onPressed: () {
              insertMahasiswa();
            },
            child: Text(
              "Simpan",
              style: TextStyle(color: Colors.white),
            ),
            color: Colors.blue,
          ),
          SizedBox(
```

```
            height: 29,
          ),
          Text("Data Mahasiswa"),
          Divider(
            thickness: 1,
          ),
          SizedBox(
            height: 10,
          ),
          listMahasiswa.length == null
              ? Center(
                  child: Text("Data Kosong"),
                )
              : ListView.builder(
                  shrinkWrap: true,
                  itemCount: listMahasiswa.length,
                  physics: ClampingScrollPhysics(),
                  itemBuilder: (c, i) {
                    return Padding(
                      padding: const EdgeInsets.all(8.0),
                      child: Column(
                        mainAxisAlignment: MainAxisAlignment.start,
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Column(
                            mainAxisAlignment: MainAxisAlignment.start,
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                              Text("Nim Mhs"),
                              Text(listMahasiswa[i].nim),
                            ],
                          ),
                          SizedBox(
                            height: 5,
                          ),
                          Column(
                            mainAxisAlignment: MainAxisAlignment.start,
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                              Text("Nama Mhs"),
                              Text(listMahasiswa[i].nama),
                            ],
                          ),
                          SizedBox(
                            height: 5,
                          ),
                          Column(
                            mainAxisAlignment: MainAxisAlignment.start,
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                              Text("Kelas Mhs"),
                              Text(listMahasiswa[i].kelas),
                            ],
                          ),
                          SizedBox(
                            height: 5,
                          ),
                          Column(
                            mainAxisAlignment: MainAxisAlignment.start,
```

```
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                              Text("Jurusan Mhs"),
                              Text(listMahasiswa[i].jurusan),
                            ],
                          ),
                          SizedBox(
                            height: 10,
                          ),
                          Divider(
                            thickness: 2,
                          ),
                        ],
                      ),
                    );
                  })
          ],
        ),
      ),
    );
  }
}
```

17. Jalankan hingga hasilnya seperti dibawah ini.

D. Daftar Pustaka

1. https://www.petanikode.com/json-pemula/
2. Modul Google Flutter Mobile Development Quick Start Guide

E. Latihan dan Tugas

Buatlah sebuah *project* seperti langkah-langkah praktikum yang sudah dilakukan diatas dengan menambahkan data untuk menyimpan nilai akhir mahasiswa yang menyimpan nilai rata-rata dari nilai akhir yang terdiri dari beberapa kriteria penilaian seperti nilai kehadiran, nilai tugas, nilai UTS dan nilai UAS.