MINGGU 10
HTTP Request

DESKRIPSI TEMA

HTTP GET & POST Request

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menerapkan fitur geolocation dan Google Map pada aplikasi Ionic
Mahasiswa mampu menghubungkan aplikasi Ionic dengan basis data online yang diakses melalui API web service.

PERALATAN YANG DIGUNAKAN

Web Storm atau VS Code, Local Web Server (jika diperlukan)

LANGKAH-LANGKAH PRAKTIKUM

Pertemuan minggu ini kita akan menggunakan local web server berisi script PHP dan juga database yang akan digunakan.

Database Setup

Buat sebuah database baru melalui phpMyAdmin, dengan nama database umn_api, dan sebuah tabel dengan nama "students"

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	nim	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
2	nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	prodi	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	foto	varchar(200)	utf8mb4_general_ci		Yes	NULL			Change Drop More

PHP Setup

1. Buat folder baru pada folder local web server Anda sehingga bisa diakses melalui http://localhost/NAMA FOLDER
2. Buat file dbconfig.php berisi code seperti berikut

```php
<?php
define('DB_USER', "root"); // db user
define('DB_PASSWORD', ""); // db password (mention your db password here)
define('DB_DATABASE', "umn_api"); // database name
define('DB_SERVER', "localhost"); // db server
?>
```

3. Buat file select_all_students.php dengan code seperti berikut

```php
<?php
if(isset($_SERVER['HTTP_ORIGIN'])){
```

```
    header('Access-Control-Allow-Origin: *');
    header('Access-Control-Allow-Methods: POST, PUT, DELETE, OPTIONS');
    header('Access-Control-Allow-Headers: Origin, X-Requested-With,
Content-Type, Accept');
}

$response = array();

// include db connect class
require_once __DIR__ . '/dbconfig.php';

// connecting to db
$db = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE) or
die(mysqli_connect_error());

$result = mysqli_query($db, "SELECT * FROM students") or
die(mysqli_connect_error());

// check for empty result
if (mysqli_num_rows($result) > 0) {
    // looping through all results
    $response["students"] = array();

    while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        // temp mahasiswa array
            $mahasiswa = array();
            $mahasiswa["nim"] = $row["nim"];
            $mahasiswa["nama"] = $row["nama"];
            $mahasiswa["prodi"] = $row["prodi"];
            $mahasiswa["foto"] = $row["foto"];
        array_push($response["students"], $mahasiswa);
    }
    // success
    $response["success"] = 1;
     echo json_encode($response);
} else {
        $response["success"] = 0;
        $response["message"] = "Tidak ada Mahasiswa yang ditemukan";
    echo json_encode($response);
}
```

```
mysqli_free_result($result);
?>
```

4. Coba buka URL untuk mengakses file PHP select_all_mahasiswa.php, jika dalam database sudah ada data, akan tampil seperti berikut.

5. Buat file insert_new_student.php dengan code seperti berikut

```
<?php
if(isset($_SERVER['HTTP_ORIGIN'])){
    header('Access-Control-Allow-Origin: *');
    header('Access-Control-Allow-Methods: POST, PUT, DELETE, OPTIONS');
    header('Access-Control-Allow-Headers: Origin, X-Requested-With,
Content-Type, Accept');
}

// array for JSON response
$response = array();

// check for required fields
if (isset($_POST['nim']) && isset($_POST['nama']) && isset($_POST['prodi']) &&
isset($_FILES['foto']) ) {

    $nim = $_POST['nim'];
```

```
    $nama = $_POST['nama'];
    $prodi = $_POST['prodi'];
    $foto = $_FILES['foto'];

    require_once __DIR__ . '/dbconfig.php';

// connecting to db
    $db = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE) or
die(mysqli_connect_error());

    // save uploaded image
    $source = $foto['tmp_name'];
    $destination = 'uploads/' . $foto['name'];
    move_uploaded_file($source, $destination);

    // mysql inserting a new row
    $result = mysqli_query($db, "INSERT INTO students(nim, nama, prodi, foto)
VALUES('$nim', '$nama', '$prodi','$destination')");

    // check if row inserted or not
    if ($result) {
        // successfully inserted into database
        $response["success"] = 1;
        $response["message"] = "Data mahasiswa berhasil dimasukkan";
    } else {
        // failed to insert row
        $response["success"] = 0;
        $response["message"] = "Ada kesalahan";
    }
     // echoing JSON response
     echo json_encode($response);
} else {
    // required field is missing
    $response["success"] = 0;
    $response["message"] = "data tidak lengkap";

    // echoing JSON response
    echo json_encode($response);
}
```

6. Buat file form_new_student.php dengan code seperti berikut

```
<form method = "POST" action = "insert_new_student.php"
enctype="multipart/form-data">
<table>
<tr> <th colspan="2">Tambah Data Mahasiswa</th></tr>
<tr> <td>NIM</td> <td><input type = "text" name = "nim"></td> </tr>
<tr> <td>NAMA</td> <td><input type = "text" name = "nama"></td> </tr>
<tr> <td>PRODI</td> <td><input type = "email" name = "prodi"></td> </tr>
<tr> <td>FOTO</td> <td><input type = "file" name = "foto"></td> </tr>
<tr> <td colspan="2"><input type = "submit" name = "submit" value = "Submit">
</td> </tr> </table>
</form>
```

7. Coba buka URL form_new_student.php melalui localhost, coba tambahkan data student, apakah berhasil menambah data ke database dan melakukan upload file foto?

Get Data with Fetch

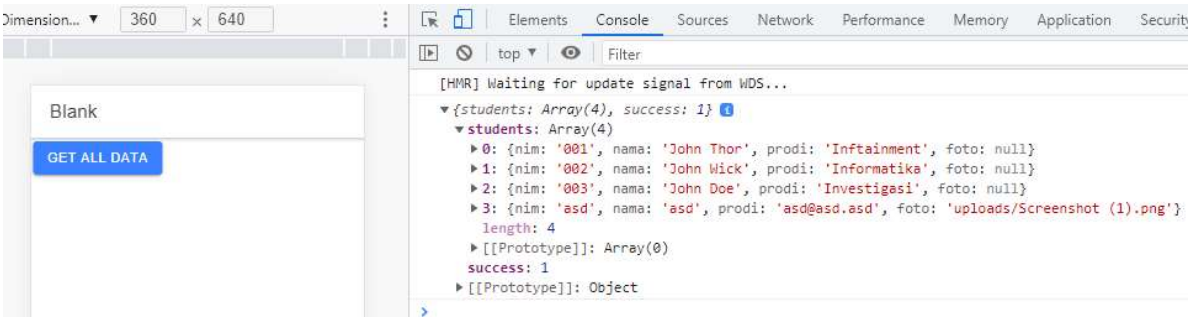
1. Buat sebuah project Ionic-React baru. Pada Home.tsx, tambahkan code berikut

```
const [data, setData] = useState( initialState: null);
const url = "http://localhost/api/select_all_students.php";

const getAllDataHandler = () => {
    fetch(url) Promise<Response>
        .then(response => response.json()) Promise<any>
        .then((data)=> {
            setData(data)
            console.log(data);
        });
};
```

```
<IonButton onClick={getAllDataHandler}>Get All Data</IonButton>
```

2. Coba hasilnya melalui browser, apakah berhasil mengambil data dari web server dan menampilkannya pada jendela console?



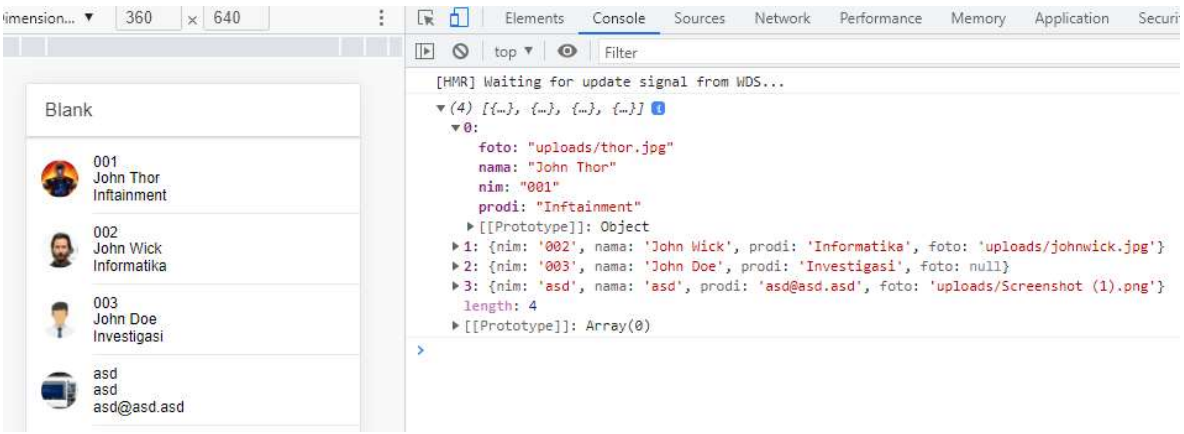
3. Untuk menampilkan data, tambahkan code berikut pada Home.tsx

```
const [data, setData] = useState<Response>();
const url = "http://localhost/api/select_all_students.php";
const [students, setStudents] = useState<Array<Student>>( initialState: []);

useEffect( effect: () => {
    fetch(url)
        .then(response => response.json())
        .then((data :Response )=> {
            setData(data)
            console.log(data.students);
            setStudents(data.students);
        });
}, deps: []);
```

```
<IonList>
    {students.map(student => (
        <IonItem key={student.nim}>
            <IonAvatar slot="start">
                <img src={"http://localhost/api/" +
                (student.foto ? student.foto : 'uploads/man.jpg')} />
            </IonAvatar>
            <IonLabel>
                {student.nim}<br />
                {student.nama}<br />
                {student.prodi}
            </IonLabel>
        </IonItem>
    ))}
</IonList>
```

4. Lihat hasilnya di browser



Get Data with Axios

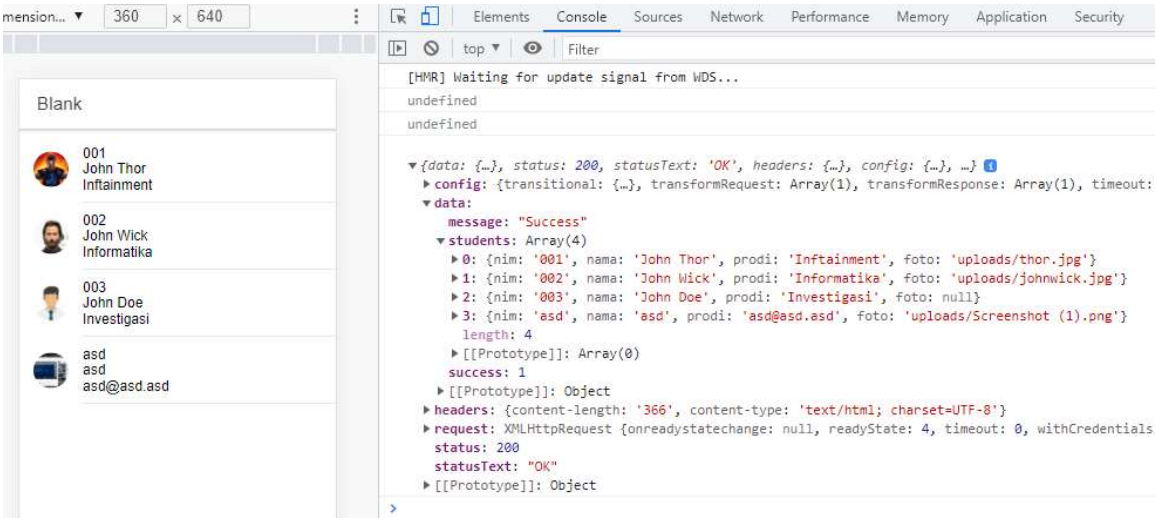
1. Install Axios dengan command: "npm install axios"
2. Tambahkan/ubah code berikut pada Home.tsx

```
const [data, setData] = useState<AxiosResponse>();
const url = "http://localhost/api/select_all_students.php";
const [students, setStudents] = useState<Array<Student>>( initialState: []);

useEffect( effect: () => {
    // fetch(url)...
    axios.get(url).then((response :AxiosResponse<any> ) => {
        setData(response);
        console.log(data);
        // setStudents(data?.data.students);
    });
}, deps: []);

useEffect( effect: () => {
    console.log(data);
    setStudents(data?.data.students);
}, deps: [data]);
```

3. Lihat hasilnya di browser



Post Data with Fetch

1. Ubah code pada Home.tsx menjadi seperti berikut

```
const [data, setData] = useState( initialState: '');
const url = "http://localhost/api/insert_new_student.php";
const nim = useRef<HTMLIonInputElement>( initialValue: null);
const nama = useRef<HTMLIonInputElement>( initialValue: null);
const prodi = useRef<HTMLIonInputElement>( initialValue: null);

const [selectedFile, setSelectedFile] = useState<File>();

const fileChangeHandler = (event: React.ChangeEvent<HTMLInputElement>) => {...};

const insertHandler = () => {...};

return (
    <IonPage...>
);
```

```
const fileChangeHandler = (event: React.ChangeEvent<HTMLInputElement>) => {
    setSelectedFile(event.target!.files![0]);
};
```

```
const insertHandler = () => {
    const formData = new FormData();

    const inNim = nim.current?.value as string;
    const inNama = nama.current?.value as string;
    const inProdi = prodi.current?.value as string;

    formData.append( name: 'nim', inNim);
    formData.append( name: 'nama', inNama);
    formData.append( name: 'prodi', inProdi);
    formData.append( name: 'foto', selectedFile as File);

    fetch(url, init: {
        method: "post",
        body:formData
    }).then(response => response.text()).then((data : string )=>{
        setData(data);
        console.log(data);
    });
};
```

```
<IonItem>
    <IonLabel position="floating">NIM</IonLabel>
    <IonInput ref={nim}></IonInput>
</IonItem>
<IonItem>
    <IonLabel position="floating">Nama</IonLabel>
    <IonInput ref={nama}></IonInput>
</IonItem>
<IonItem>
    <IonLabel position="floating">Email</IonLabel>
    <IonInput ref={prodi}></IonInput>
</IonItem>
<IonItem>
    <input type="file" onChange={fileChangeHandler} />
</IonItem>
<IonButton onClick={insertHandler}>Simpan</IonButton>
```

2. Coba jalankan di browser dan coba tambahkan mahasiswa baru, apakah data berhasil disimpan ke database dan foto terupload?

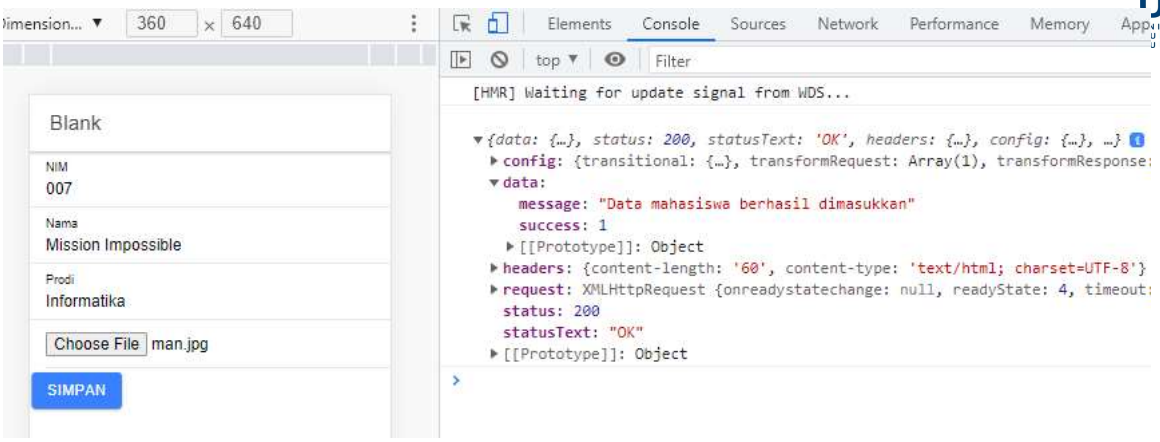
Post Data with Axios

1. Ubah code pada Home.tsx menjadi seperti berikut

```
// fetch(url, {
//     method: "post",
//     body:formData
// }).then(response => response.text()).then((data)=>{
//     setData(data);
//     console.log(data);
// });

axios.post(url, formData).then(res => {
    console.log(res);
});
```

2. Coba jalankan di browser dan tambah data mahasiswa baru, apakah berhasil upload file dan data tersimpan di database?



TUGAS

1. Buat API sederhana menggunakan PHP dan MySQL untuk menyimpan dan mengambil data Memories (project pertemuan sebelumnya)
2. Modifikasi project Memories minggu lalu, sehingga data Good dan Bad Memory tersimpan di database MySQL