

Nama: Farhan Hibatullah

NPM: 1915061014

Tentang: Tugas Akhir Praktikum Pemrograman Web (AJAX)

Soal:

Gabungkan seluruh kode Ajax dalam 1 berkas html. Lakukan modifikasi yang diperlukan agar PUT dan DELETE tidak lagi menerima input dari tag <input> tetapi harus menggunakan ID dari hidden input yang disematkan pada tabel yang dihasilkan oleh GET. Dan, pastikan berfungsi!

Jawab:

Disini saya menggunakan api.php yang sudah dibuat kemudian menggabungkannya menjadi satu file html, dimana saya juga membuat file javascript eksternal untuk memudahkan dalam pengembangan sistem CRUD yang diinginkan.

Source Code:

1. File HTML (index.html)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>AJAX CRUD</title>
</head>
<body>
  <h1>AJAX CRUD</h1>
  <h2>GET</h2>
  <button id="btnAmbilData">Ambil Data</button>
  <div id="tampilanData"></div>
  <div id="editData"></div>

  <h2>POST</h2>
  <input type="text" name="nama" id="nama" placeholder="Ketik nama disini">
  <input type="text" name="npm" id="npm" placeholder="Ketik NPM disini">
  <button id="btnKirimData">Kirim Data</button>
  <div id="tampilanDataPost"></div>

  <script src="script.js"></script>
</body>
</html>
```

2. File Javascript (script.js)

```
var tampilanData = document.getElementById("tampilanData");
var tampilanDataPost = document.getElementById("tampilanDataPost");

function AjaxGET() {
    var data_file = "api.php/mahasiswa?t=" + Math.random();
    var http_request = new XMLHttpRequest();
    try {
        http_request = new XMLHttpRequest();
    } catch (e) {
        try {
            http_request = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {
                alert("Browser rusak!");
                return false;
            }
        }
    }
}

http_request.onreadystatechange = function() {
    if (http_request.readyState == 4) {
        var jsonObj = JSON.parse(http_request.responseText);
        tampilanData(jsonObj);
    }
}

http_request.open("GET", data_file, true);
http_request.send();

function createNode(element){
    return document.createElement(element);
}

function append(parent, el){
    return parent.appendChild(el);
}

function tampilanData(dataRAW) {
    tampilanData.innerHTML = "Status: " + dataRAW.status;
    let table = createNode('table'),
    thead = createNode('thead'),
    th1 = createNode('th'),
    th2 = createNode('th'),
    thEdit = createNode('th'),
    thDelete = createNode('th');
```

```

th1.innerHTML = 'Nama';
th2.innerHTML = 'NPM';
thEdit.innerHTML = 'Aksi1';
thDelete.innerHTML = 'Aksi2';

append(thead, th1);
append(thead, th2);
append(thead, thEdit);
append(thead, thDelete);
append(table, thead);

for (let i = 0; i < dataRAW.data.length; i++) {
    let row = createNode('tr'),
        td1 = createNode('td'),
        td2 = createNode('td'),
        tdEdit = createNode('td'),
        tdDelete = createNode('td'),
        editBtn = createNode('button'),
        deleteBtn = createNode('button');

    td1.innerHTML = dataRAW.data[i].nama;
    td2.innerHTML = dataRAW.data[i].npm;
    editBtn.innerHTML = 'Edit';
    deleteBtn.innerHTML = 'Delete';

    editBtn.onclick = function () {
        editData(dataRAW.data[i].id,
dataRAW.data[i].nama, dataRAW.data[i].npm);
    }

    deleteBtn.onclick = function () {
        deleteData(dataRAW.data[i].id);
    }

    append(tdEdit, editBtn);
    append(tdDelete, deleteBtn);
    append(row, td1);
    append(row, td2);
    append(row, tdEdit);
    append(row, tdDelete);
    append(table, row);
}

append(tampilanData, table);
}
}

document.getElementById("btnAmbilData").addEventListener("click", AjaxGET);

```

```

function AjaxPOST() {
    var nama = document.getElementById("nama").value;
    var npm = document.getElementById("npm").value;
    var data_file = "api.php/mahasiswa";
    var params = "nama=" + nama + "&npm=" + npm;
    var http_request = new XMLHttpRequest();
    try {
        http_request = new XMLHttpRequest();
    } catch (e) {
        try {
            http_request = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {
                alert("Browser rusak!");
                return false;
            }
        }
    }
    http_request.onreadystatechange = function () {
        if (http_request.readyState == 4) {
            var jsonObj = JSON.parse(http_request.responseText);
            tampilkanDataPost(jsonObj);
        }
    }
    http_request.open("POST", data_file, true);
    http_request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    http_request.send(params);

    function tampilkanDataPost(dataRAW) {
        tampilkanDataPost.innerHTML = "Status: " + dataRAW.status + "<br/>";
        tampilkanDataPost.innerHTML += "ID baru: " + dataRAW.data.id +
"<br/>";
        tampilkanDataPost.innerHTML += "Nama: " + dataRAW.data.nama +
"<br/>";
        tampilkanDataPost.innerHTML += "NPM: " + dataRAW.data.npm + "<br/>";
    }
}

document.getElementById("btnKirimData").addEventListener("click", AjaxPOST);

function editData(id, nama, npm) {
    var tampilEdit = document.getElementById("editData");
    let inputNama = document.createElement("input"),
    inputNPM = document.createElement("input"),

```

```

submit = document.createElement("button");

inputNama.setAttribute("type", "text");
inputNama.setAttribute("id", "nama");
inputNama.setAttribute("value", nama);

inputNPM.setAttribute("type", "text");
inputNPM.setAttribute("id", "npm");
inputNPM.setAttribute("value", npm)

submit.innerHTML = "Submit"
submit.onclick = function(){
    var nama = document.getElementById("nama").value;
    var npm = document.getElementById("npm").value;
    var data_file = "api.php/mahasiswa/" + id;
    var http_request = new XMLHttpRequest();
    try {
        http_request = new XMLHttpRequest();
    } catch (e) {
        try {
            http_request = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {
                alert("Browser rusak!");
                return false;
            }
        }
    }
    http_request.onreadystatechange = function () {
        if (http_request.readyState == 4) {
            var jsonObj = JSON.parse(http_request.responseText);
            tampilkanDataPut(jsonObj);
        }
    }

    http_request.open("PUT", data_file, true);
    http_request.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    http_request.send("id=" + id + "&nama=" + nama + "&npm=" + npm);

    function tampilkanDataPut(dataRAW) {
        tampilEdit.innerHTML = "Status: " + dataRAW.status + "<br/>";
        tampilEdit.innerHTML += "ID yang diubah: " + dataRAW.data.id +
"<br/>";
        tampilEdit.innerHTML += "Nama baru: " + dataRAW.data.nama +
"<br/>";
    }
}

```

```

        tampilEdit.innerHTML += "NPM baru: " + dataRAW.data.npm +
"<br/>";
    }
}

tampilEdit.appendChild(inputNama);
tampilEdit.appendChild(inputNPM);
tampilEdit.appendChild(submit);
}

function deleteData(id) {
    var data_file = "api.php/mahasiswa/" + id;
    var http_request = new XMLHttpRequest();
    try {
        http_request = new XMLHttpRequest();
    } catch (e) {
        try {
            http_request = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (e) {
                alert("Browser rusak!");
                return false;
            }
        }
    }
}

http_request.onreadystatechange = function () {
    if (http_request.readyState == 4) {
        var jsonObj = JSON.parse(http_request.responseText);
        tampilkanDataDelete(jsonObj);
    }
}

http_request.open("DELETE", data_file, true);
http_request.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
http_request.send();

function tampilkanDataDelete(dataRAW) {
    tampilData.innerHTML = "Status: " + dataRAW.status + "<br/>";
    tampilData.innerHTML += "ID yang dihapus: " + dataRAW.data.id +
"<br/>";
    tampilData.innerHTML += "Nama: " + dataRAW.data.nama + "<br/>";
    tampilData.innerHTML += "NPM: " + dataRAW.data.npm + "<br/>";
}
}

```

3. File PHP (api.php)

```
<?php
header('Content-type: application/json');
$method = $_SERVER['REQUEST_METHOD'];

$request = explode("/", substr(@$_SERVER['PATH_INFO'], 1));
function process_get($param){
    if ($param[0] == "mahasiswa"){
        require_once 'dbconfig.php';
        try {
            $conn = new PDO(
                "mysql:host=$host;dbname=$dbname",
                $username,
                $password,
                array(
                    \PDO::ATTR_ERRMODE =>
                    \PDO::ERRMODE_EXCEPTION,
                    \PDO::ATTR_PERSISTENT => false
                )
            );

            if (!empty($param[1])) {
                $handle = $conn->prepare(" SELECT id, nama, npm FROM
mahasiswa WHERE id = :id ");
                $handle->bindParam(':id', $param[1], PDO::PARAM_INT);
                $handle->execute();
            } else {
                $handle = $conn->query("SELECT id, nama, npm from
mahasiswa");
            }
            if ($handle->rowCount()) {
                $status = 'Berhasil';
                $data = $handle->fetchAll(PDO::FETCH_ASSOC);
                $arr = array('status' => $status, 'data' => $data);
            } else {
                $status = "Tidak ada data";
                $arr = array('status' => $status);
            }
            echo json_encode($arr);
        } catch (PDOException $pe) {
            die(json_encode($pe->getMessage()));
        }
    } else{
    }
}

function process_post($param){
    if ((count($param) == 1) and ($param[0] == "mahasiswa")){
```

```

require_once 'dbconfig.php';
$dataNama = (isset($_POST['nama']) ? $_POST['nama'] : NULL);
$dataNPM = (isset($_POST['npm']) ? $_POST['npm'] : NULL);
try {
    $conn = new PDO(
        "mysql:host=$host;dbname=$dbname",
        $username,
        $password,
        array(
            \PDO::ATTR_ERRMODE =>
            \PDO::ERRMODE_EXCEPTION,
            \PDO::ATTR_PERSISTENT => false
        )
    );
    $handle = $conn->prepare("INSERT INTO mahasiswa (nama, npm)
VALUES (:nama, :npm)");
    $handle->bindParam(':nama', $dataNama);
    $handle->bindParam(':npm', $dataNPM);
    $handle->execute();
    if ($handle->rowCount()) {
        $status = 'Berhasil';
        $idTerakhir = $conn->lastInsertId();
        $arr = array('status' => $status, 'id' => $idTerakhir);
    } else {
        $status = "Gagal";
        $arr = array('status' => $status);
    }
    echo json_encode($arr);
} catch (PDOException $pe) {
    die(json_encode($pe->getMessage()));
}
} else {
    echo "param kelebihan atau parameter bukan mahasiswa atau tidak
ada parameter";
}
}

function process_put($param){
    if ((count($param) == 2) and $param[0] == "mahasiswa" and
        $_SERVER["CONTENT_TYPE"] == 'application/x-www-form-urlencoded'
    ) {
        require_once 'dbconfig.php';
        parse_str(file_get_contents('php://input'), $data);
        $dataNama = $data['nama'];
        $dataNPM = $data['npm'];
        try {
            $conn = new PDO(
                "mysql:host=$host;dbname=$dbname",

```



```

        $username,
        $password,
        array(
            \PDO::ATTR_ERRMODE =>
            \PDO::ERRMODE_EXCEPTION,
            \PDO::ATTR_PERSISTENT => false
        )
    );
    $handle = $conn->prepare("UPDATE mahasiswa SET nama=:nama,
npm=:npm, tanggal_tercatat=NOW() WHERE ID=:id");
    $dataID = $param[1];
    $handle->bindParam(':id', $dataID, PDO::PARAM_INT);
    $handle->bindParam(':nama', $dataNama);
    $handle->bindParam(':npm', $dataNPM);
    $handle->execute();
    if ($handle->rowCount()) {
        $status = 'Berhasil';
    } else {
        $status = "Gagal";
    }
    $arr = array('status' => $status, 'id' => $dataID, 'nama'
=>$dataNama, 'npm' => $dataNPM);
    echo json_encode($arr);
} catch (PDOException $pe) {
    die(json_encode($pe->getMessage()));
}
}

function process_delete($param){
    if ((count($param) == 2) and $param[0] == "mahasiswa") {
        require_once 'dbconfig.php';
        try {
            $conn = new PDO(
                "mysql:host=$host;dbname=$dbname",
                $username,
                $password,
                array(
                    \PDO::ATTR_ERRMODE =>
                    \PDO::ERRMODE_EXCEPTION,
                    \PDO::ATTR_PERSISTENT => false
                )
            );
            $handle = $conn->prepare(" DELETE FROM mahasiswa WHERE
ID=:id ");

            $dataID = $param[1];
            $handle->bindParam(':id', $dataID, PDO::PARAM_INT);
            $handle->execute();

```

```

        if ($handle->rowCount()) {
            $status = 'Berhasil';
        } else {
            $status = "Gagal";
        }
        $arr = array('status' => $status, 'id' => $dataID);
        echo json_encode($arr);
    } catch (PDOException $pe) {
        die(json_encode($pe->getMessage()));
    }
}

function handle_error($param){
    if($param != 'mahasiswa' and $param == null){
        die("Kesalahan Parameter!");
    }
}

switch ($method) {
    case 'PUT':
        process_put($request);
        break;
    case 'POST':
        process_post($request);
        break;
    case 'GET':
        process_get($request);
        break;
    case 'DELETE':
        process_delete($request);
        break;
    default:
        handle_error($request);
        break;
}
?>

```

4. File PHP (dbconfig.php)

```

<?php
$host = 'localhost';
$dbname = 'kuliah';
$username = 'root';
$password = '';
?>

```

5. File PHP (buatable.php)

```
<?php
require_once 'dbconfig.php';
try {
    $conn = new PDO(
        "mysql:host=$host;dbname=$dbname",
        $username,
        $password,
        array(
            \PDO::ATTR_ERRMODE => \PDO::ERRMODE_EXCEPTION,
            \PDO::ATTR_PERSISTENT => false
        )
    );
    $handle = $conn->prepare("
CREATE TABLE mahasiswa (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nama VARCHAR(30) NOT NULL,
npm VARCHAR(10) NOT NULL,
tanggal_tercatat TIMESTAMP
)
");
    $handle->execute();
    echo "Tabel berhasil dibuat.";
} catch (PDOException $pe) {
    die("Tabel gagal dibuat: " . $pe->getMessage());
}
?>
```

Output:

AJAX CRUD

GET

Ambil Data

POST

Ketik nama disini Ketik NPM disini Kirim Data

Gambar 1. Tampilan index.html

Berdasarkan Gambar 1, ditampilkan sebuah halaman index.html dengan judul AJAX CRUD dimana terdapat GET yang diikuti sebuah button Ambil Data, dan POST yang diikuti dengan input kolom untuk nama dan NPM beserta satu buah button Kirim Data.

AJAX CRUD

GET

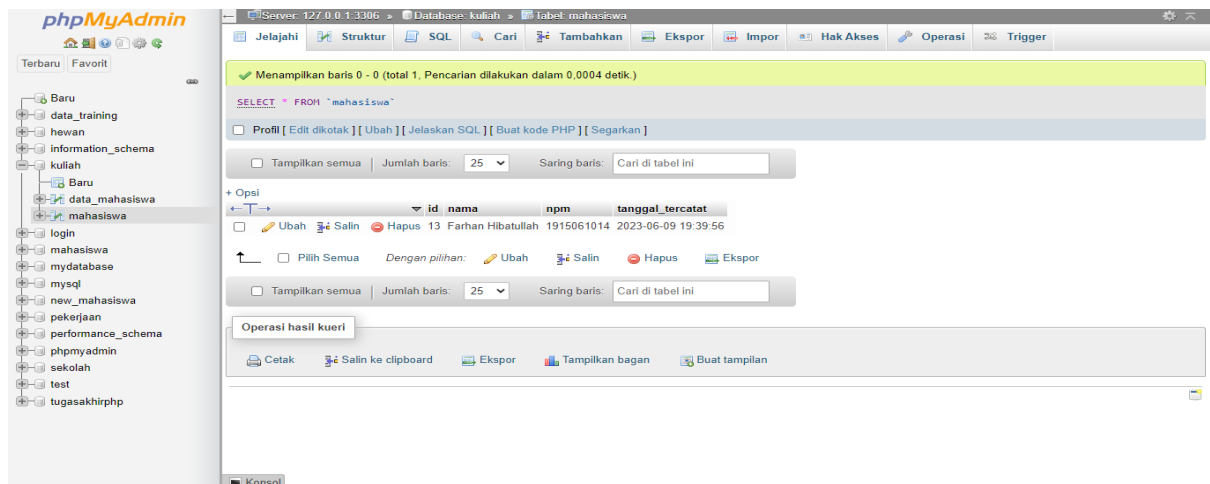
Ambil Data

POST

Farhan Hibatullah 1915061014 Kirim Data
Status: Berhasil

Gambar 2. Tampilan POST Data Berhasil

Berdasarkan Gambar 2, terdapat tampilan berupa tiap kolom input yang telah diisi, input nama berupa Farhan Hibatullah, dan input NPM berupa 1915061014, kemudian dilakukan aksi mengirim data ke server menggunakan button Kirim Data dengan metode POST, lalu diakhiri tampilan status ketika permintaan pengiriman data menuju server telah berhasil yaitu “Status: Berhasil”.



Gambar 3. Tampilan Database Ketika POST Data Berhasil

Berdasarkan Gambar 3, terdapat tampilan database phpMyAdmin yang diakses melalui localhost yang disediakan oleh XAMPP, dimana ditampilkan tabel mahasiswa, yang berisi id dengan nilai 13, nama dengan nilai Farhan Hibatullah, npm dengan nilai 1915061014, dan tanggal_tercatat dengan nilai timestamp ketika data tersebut diinsert. Hal ini berarti proses pengiriman data menggunakan api.php dengan metode POST telah berhasil dijalankan.

AJAX CRUD

GET

Ambil Data
 Status: Berhasil

Nama	NPM	Aksi1	Aksi2
Farhan Hibatullah	1915061014	Edit	Delete

POST

Ketik nama disini Ketik NPM disini Kirim Data

Gambar 4. Tampilan GET Data Berhasil

Berdasarkan Gambar 4, terdapat tampilan tabel yang berisi Nama dengan nilai Farhan Hibatullah, NPM dengan nilai 1915061014, Aksi1 yang berisi button dengan nilai Edit, dan Aksi2 yang berisi button dengan nilai Delete. Data tersebut berhasil diambil dari database phpMyAdmin dengan menjalankan script pada file api.php dengan menggunakan metode GET dimana ditunjukkan dengan munculnya tampilan status yaitu "Status: Berhasil".

AJAX CRUD

GET

Status: Berhasil

Nama	NPM	Aksi1	Aksi2
Farhan Hibatullah	1915061014	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
<input type="text" value="Farhan Hibatullah"/>	<input type="text" value="1915061014"/>	<input type="button" value="Submit"/>	

POST

Gambar 5. Tampilan Aksi EDIT

Berdasarkan Gambar 5, terdapat tampilan yang menampilkan kolom input nama dan npm yang terisi masing masing dengan nilai Farhan Hibatullah dan 1915061014 serta button submit, dimana kolom input dan button ini muncul ketika button Edit diklik.

AJAX CRUD

GET

Status: Berhasil

Nama	NPM	Aksi1	Aksi2
Farhan Hibatullah	1915061014	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
<input type="text" value="Hania Fahrani"/>	<input type="text" value="1917041014"/>	<input type="button" value="Submit"/>	

POST

Gambar 6. Mengganti Nilai

Berdasarkan Gambar 6, terdapat tampilan kolom input yang telah diganti nilainya dengan masing masing nilai input berupa Hania Fahrani pada kolom input nama dan 1917041014 pada kolom input npm.

AJAX CRUD

GET

Status: Berhasil

Nama	NPM	Aksi1	Aksi2
Farhan Hibatullah	1915061014	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Status: Berhasil

POST

Gambar 7. Tampilan Submit Data

Berdasarkan Gambar 7, terdapat tampilan status berhasil ketika button submit diklik.

AJAX CRUD

GET

Status: Berhasil

Nama	NPM	Aksi1	Aksi2
Hania Fahrani	1917041014	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

POST

Gambar 8. Tampilan Setelah Edit Data

Berdasarkan Gambar 8, terdapat tampilan tabel yang diambil menggunakan metode GET dengan menjalankan script api.php dengan nilai yang sudah diganti. Hal ini berarti, metode PUT pada api.php telah berhasil dijalankan.

AJAX CRUD

GET

Status: Data Berhasil Dihapus!

POST

Gambar 9. Tampilan Tombol DELETE Data

Berdasarkan Gambar 9, terdapat tampilan status “Status: Data Berhasil Dihapus!” yang tampil setelah button Delete pada baris tabel yang bersesuaian diklik.

AJAX CRUD

GET

Status: Tidak ada data

POST

Gambar 10. Tampilan Setelah DELETE Data Berhasil

Berdasarkan Gambar 10, terdapat tampilan status “Status: Tidak ada data” yang tampil ketika dilakukan pengambilan kembali data pada database menggunakan metode GET yang dijalankan melalui file api.php dimana hal ini mengindikasikan bahwa metode DELETE pada file api.php telah berhasil dijalankan. Kesimpulannya sistem CRUD dengan menggunakan teknik AJAX telah berhasil digabungkan dan berjalan dengan baik!