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>

1. 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);

            tampilkanData(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 tampilkanData(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) {

        tampilanDataPost.innerHTML = "Status: " + dataRAW.status + "<br/>";

        tampilanDataPost.innerHTML += "ID baru: " + dataRAW.data.id + "<br/>";

        tampilanDataPost.innerHTML += "Nama: " + dataRAW.data.nama + "<br/>";

        tampilanDataPost.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) {

        tampilanData.innerHTML = "Status: " + dataRAW.status + "<br/>";

        tampilanData.innerHTML += "ID yang dihapus: " + dataRAW.data.id + "<br/>";

        tampilanData.innerHTML += "Nama: " + dataRAW.data.nama + "<br/>";

        tampilanData.innerHTML += "NPM: " + dataRAW.data.npm + "<br/>";

    }

}

1. 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;

    }

?>

1. File PHP (dbconfig.php)

<?php

$host = 'localhost';

$dbname = 'kuliah';

$username = 'root';

$password = '';

?>

1. File PHP (buattable.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.";

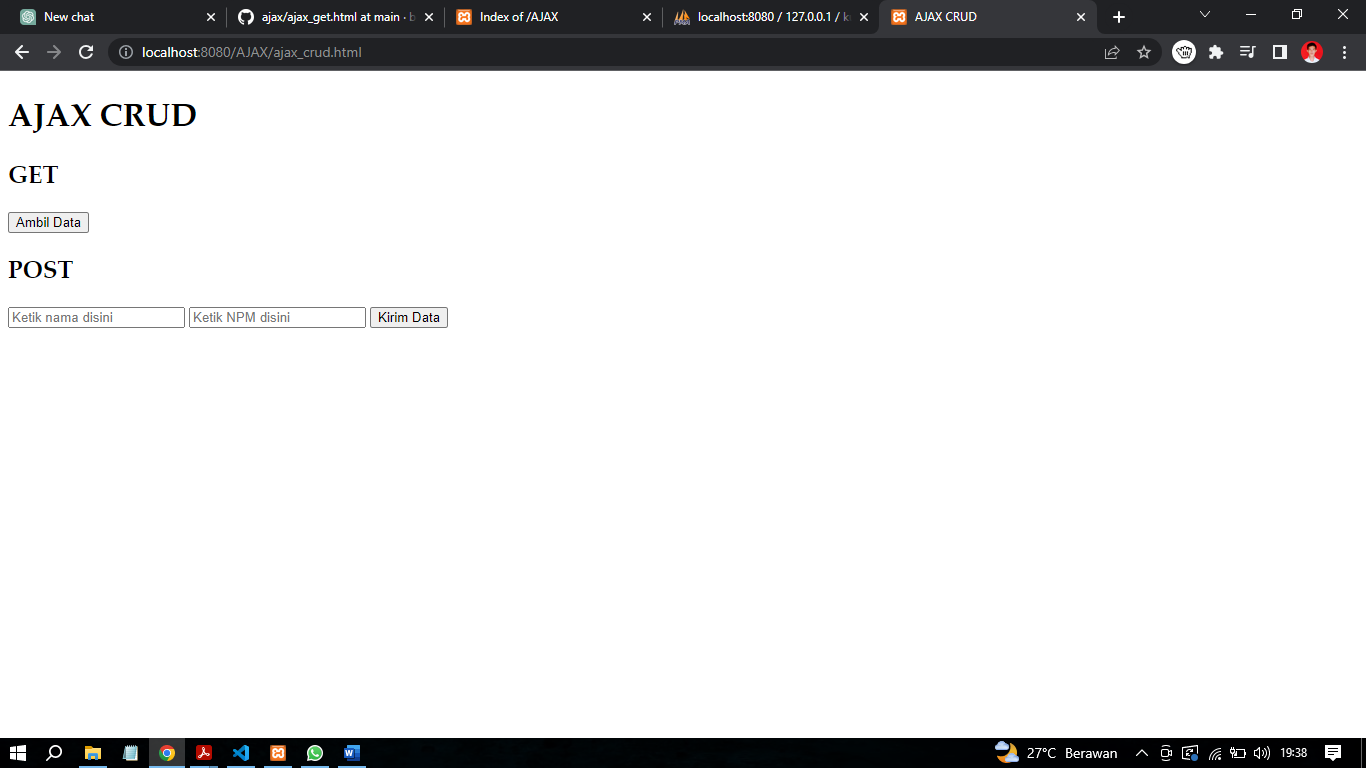
} catch (PDOException $pe) {

    die("Tabel gagal dibuat: " . $pe->getMessage());

}

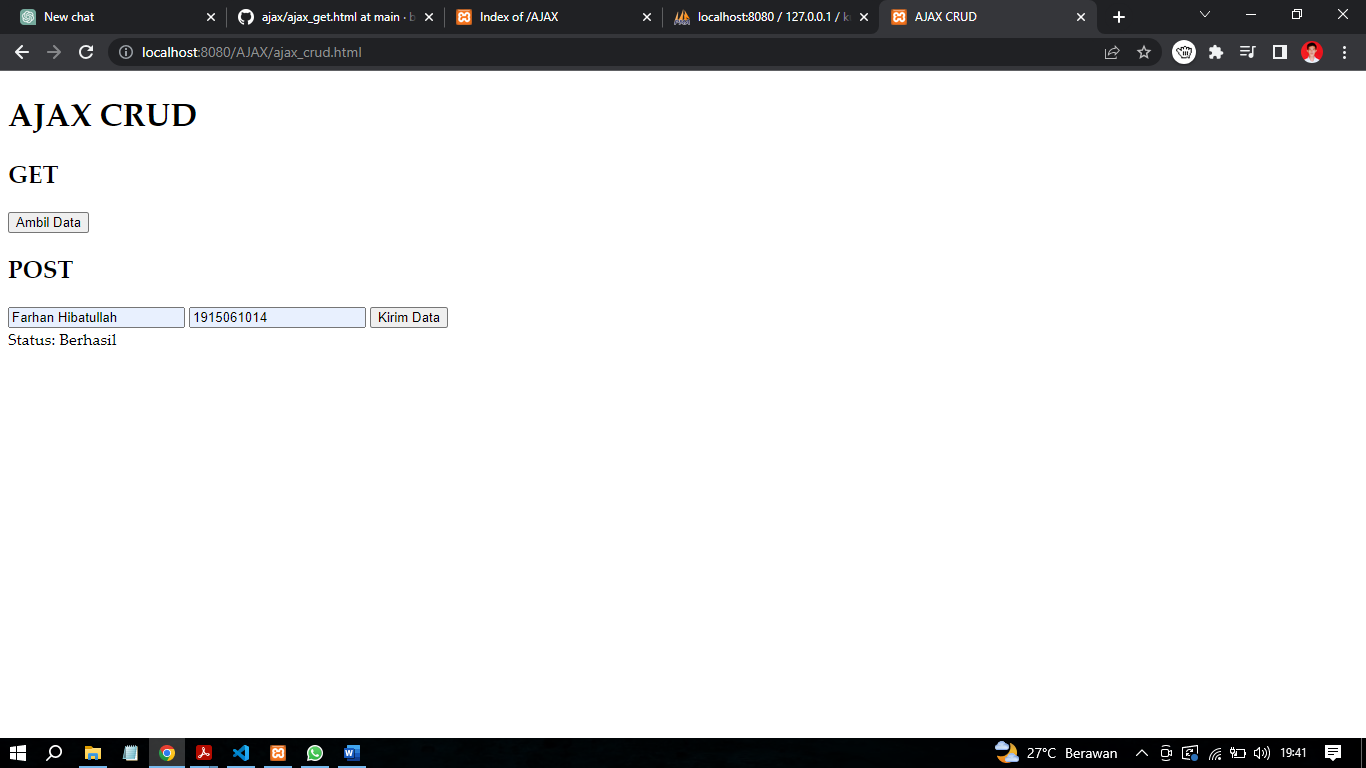
?>

Output:



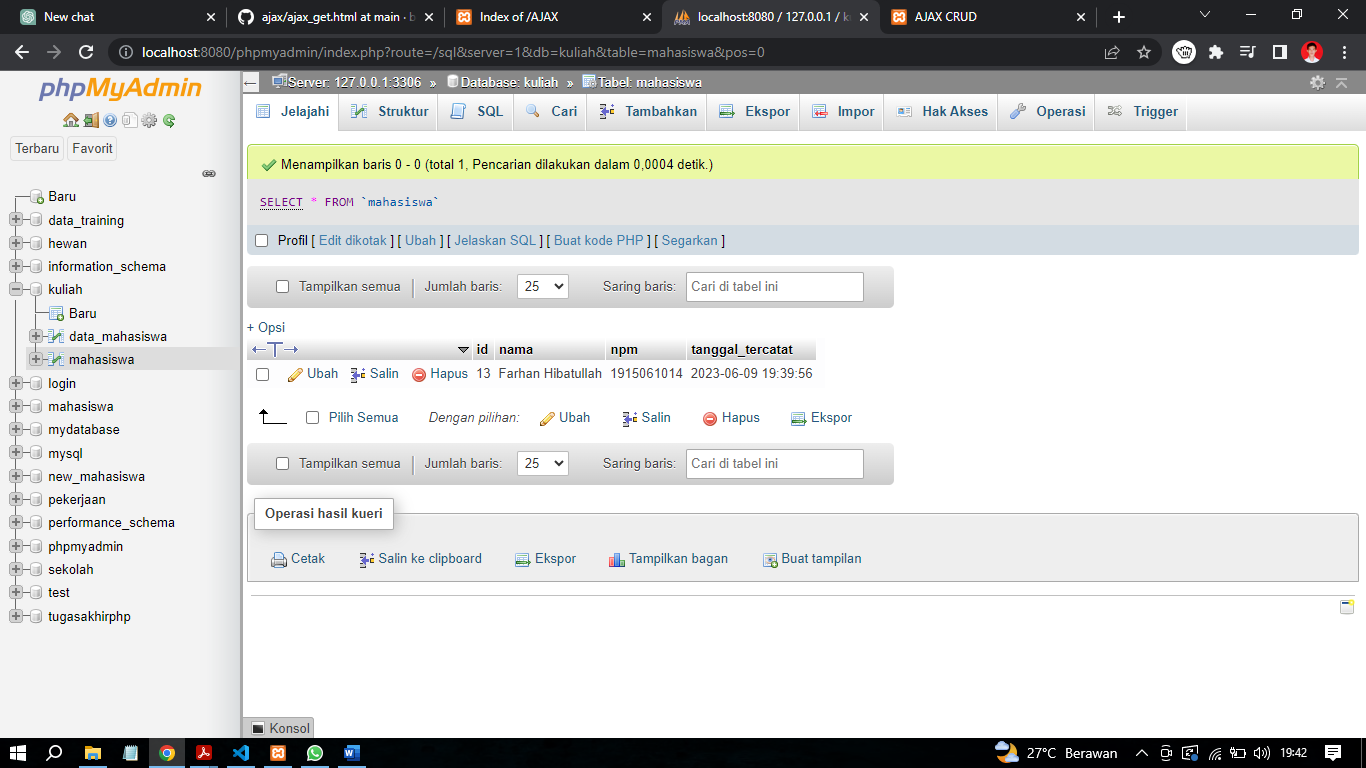
Gambar 1. Tampilan index.html

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.



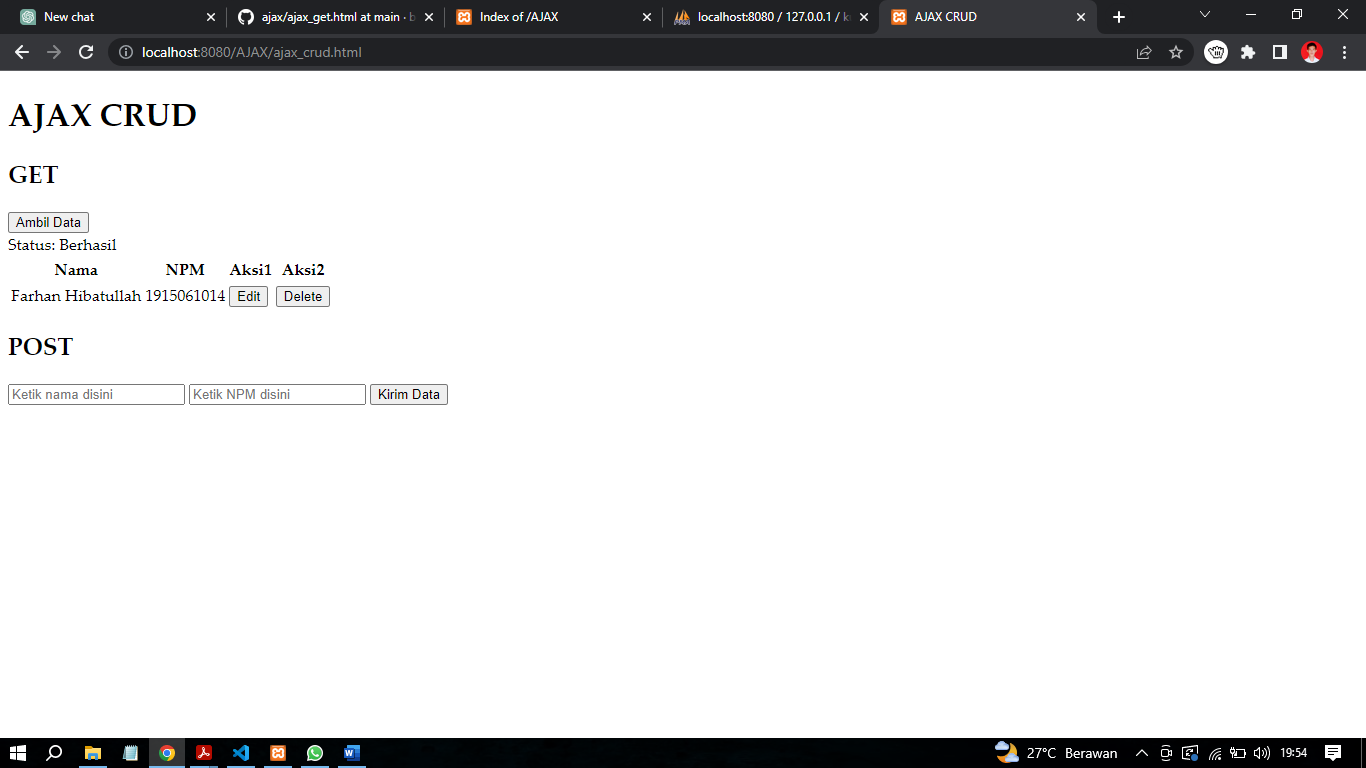
Gambar 2. Tampilan POST Data Berhasil

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”.



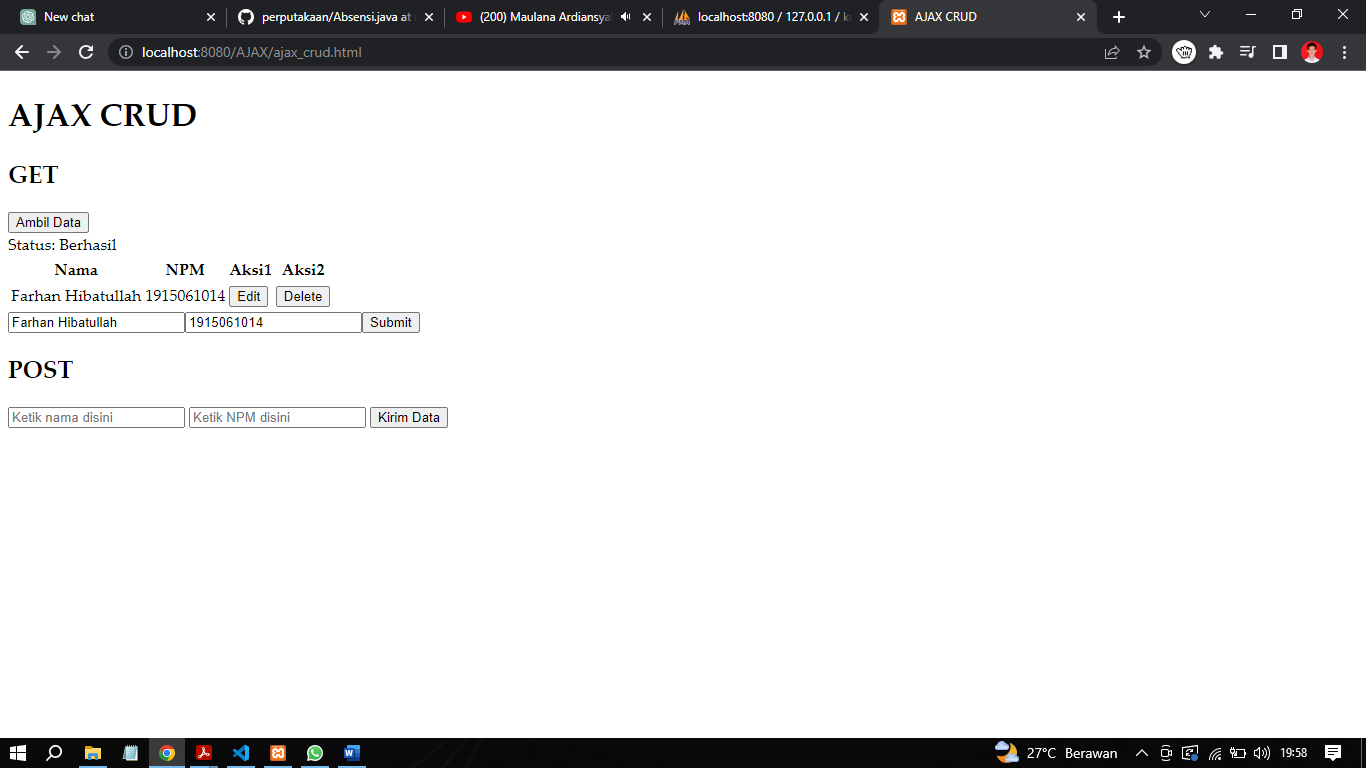
Gambar 3. Tampilan Database Ketika POST Data Berhasil

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.



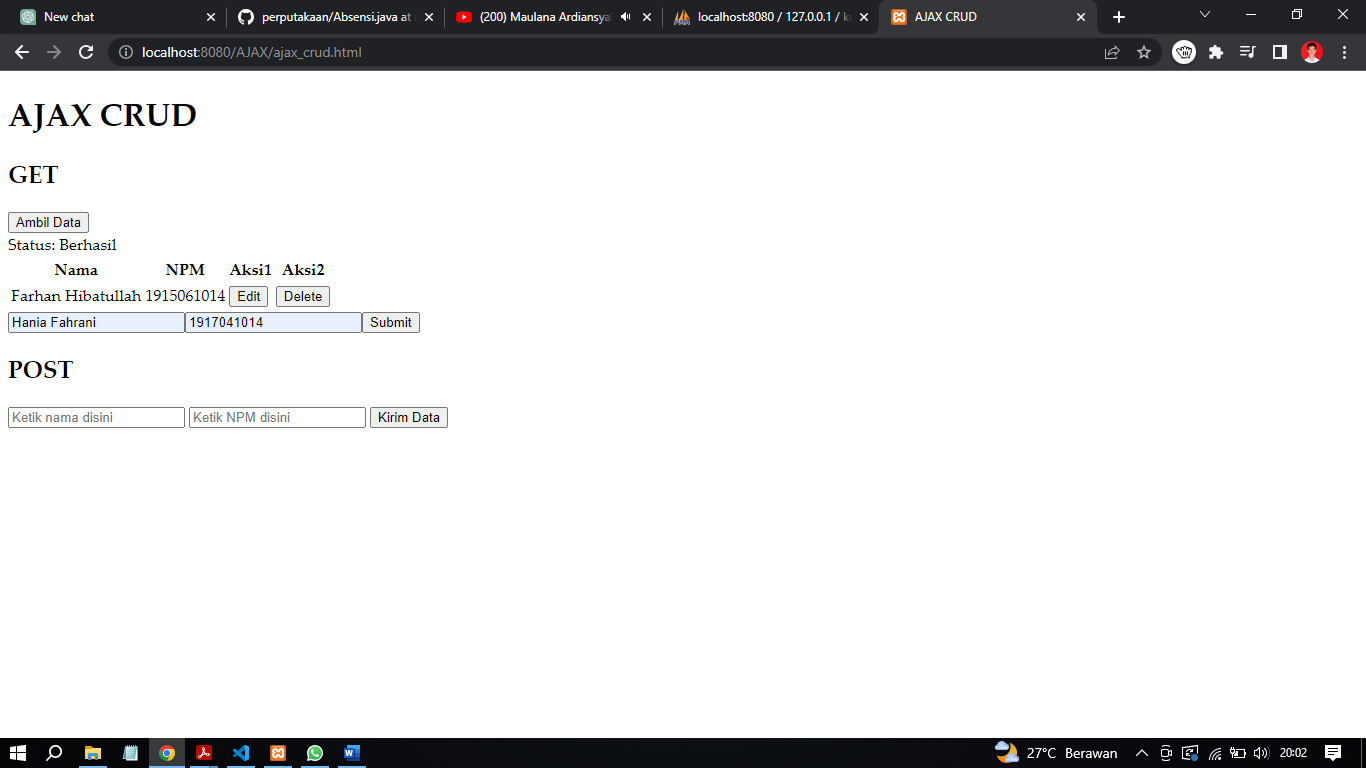
Gambar 4. Tampilan GET Data Berhasil

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”.



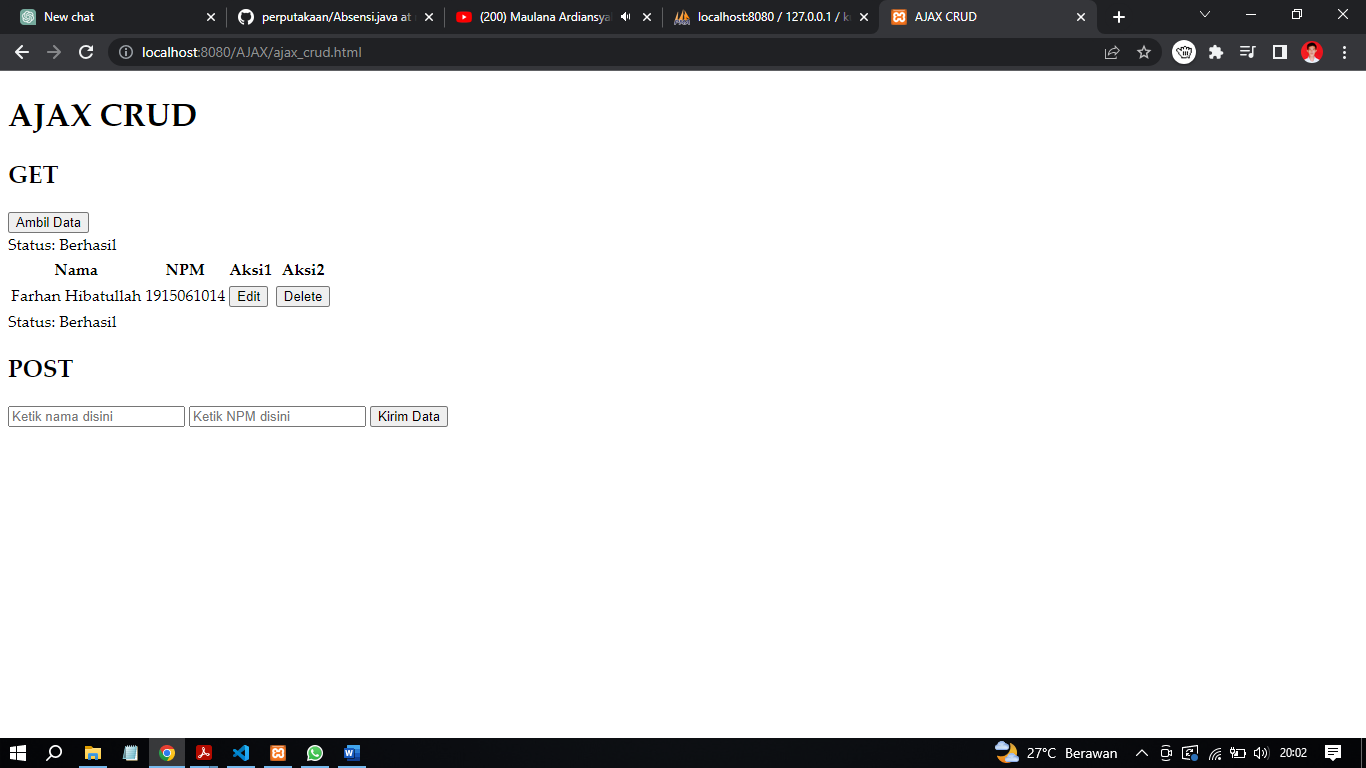
Gambar 5. Tampilan Aksi EDIT

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.



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.



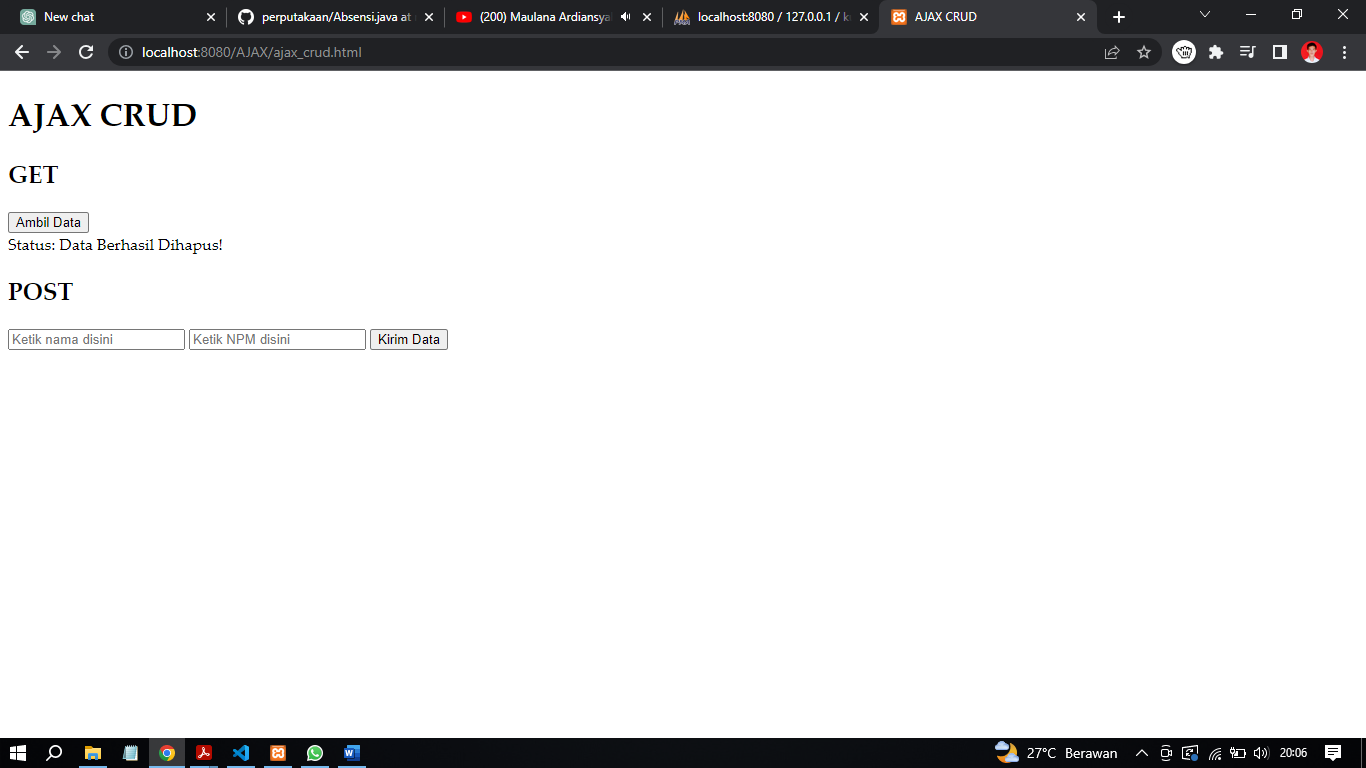
Gambar 7. Tampilan Submit Data

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



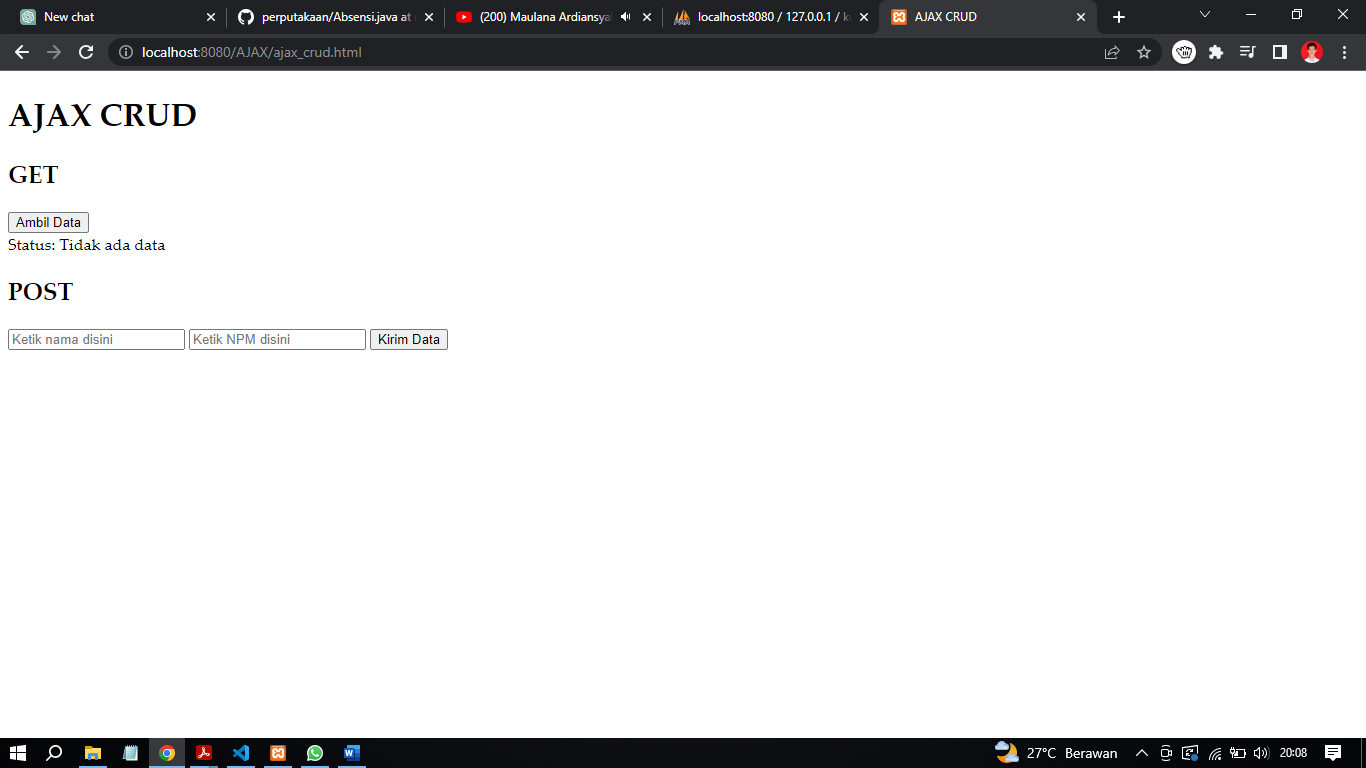
Gambar 8. Tampilan Setelah Edit Data

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.



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.



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!