

## **LAPORAN HASIL PRAKTIKUM**

### **PEMROGRAMAN WEBSITE DAN MOBILE I**



**NAMA : MUHAMMAD ARIFIN**

**NIM : 193020503025**

**KELAS : A**

**MODUL : III**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2021**

## BAB I

### TUJUAN DAN LANDASAN TEORI

#### 1. Tujuan Praktikum

- 1.1. Mahasiswa mampu membuat program yang bisa menyimpan data dalam jumlah yang banyak.
- 1.2. Mahasiswa mampu membuat program yang bisa mengolah data yang tersimpan dalam database.

#### 2. Landasan Teori

Data dalam database MySQL disimpan dalam tabel-tabel. Sebuah tabel adalah koleksi dari data yang berelasi dan mengandung kolom dan baris. Database sangat bermanfaat untuk menyimpan informasi secara kategori. Contoh yang akan diberikan pada modul praktikum ini adalah tabel yang mengandung data Employees (Pekerja), Products (Produk), Customers (Pelanggan) dan Orders (Pesanan).

##### Membuka Koneksi

Sebelum mengakses data dalam database MySQL, kita harus terhubung ke server database MySQL. Berikut adalah contoh kode program agar terhubung dengan server mySQL:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Membuat hubungan
    $conn = new mysqli($servername, $username, $password);

    // Memeriksa hubungan
    if ($conn->connect_error){
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

**Gambar 1.1** Koneksi

Jika kode diatas tidak berhasil, kemungkinan variabel \$connect\_error sudah tidak tersedia pada versi PHP yang kita gunakan, maka gunakan kode program berikut:

```
<?php

$servername = "localhost";
$username = "username";
$password = "password";

// Membuat Hubungan
$conn = mysqli_connect($servername, $username, $password);

// Memeriksa
Hubungan if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";

?>
```

**Gambar 1.2** Koneksi

Ketika selesai menggunakan data dari database, sebaiknya koneksi atau hubungan ke server ditutup, caranya dengan menggunakan kode program berikut:  
**mysqli\_close(\$conn);**

### **Membuat Database**

Database pada MySQL bisa juga dibuat menggunakan kode program PHP. Program tersebut akan berisi statement SQL “CREATE DATABASE”. Berikut adalah contoh program yang digunakan untuk membuat database “myDB”:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

// Membuat Database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {

```

**Gambar 1,3** Membuat Database

```

        echo "Database created successfully";
    } else {
        echo "Error creating database: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

**Gambar 1.4** Membuat Database

### Membuat Tabel

Pembuatan tabel pada bahasa pemrograman PHP juga menggunakan statement SQL, yaitu statement “CREATE TABLE”. Contoh yang akan diberikan adalah pembuatan tabel MyGuests. Statement pembuatan tabel MyGuests adalah sebagai berikut:

```

CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, firstname
VARCHAR(30) NOT NULL, lastname VARCHAR(30) NOT NULL, email
VARCHAR(50), reg_date TIMESTAMP )

```

Berikut adalah contoh program untuk membuat tabel diatas:

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    // SQL untuk membuat tabel
    $sql = "CREATE TABLE MyGuests (

```

**Gambar 1.5** Membuat Tabel

```

        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY
        KEY, firstname VARCHAR(30) NOT NULL, lastname
        VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP
    );

    if (mysqli_query($conn, $sql)) {
        echo "Table MyGuests created successfully";
    } else {
        echo "Error creating table: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>

```

**Gambar 1.6** Membuat Tabel

### Memasukkan Data ke Database

Untuk memasukkan data kedalam tabel di database, terdapat beberapa aturan sintaks yang harus diikuti:

- Query SQL harus diberikan kutip dalam PHP.
- Nilai string didalam query SQL harus diberikan kutip.
- Nilai numeris tidak harus diberikan kutip.
- Nilai NULL tidak harus diberikan kutip.

Statement INSERT INTO digunakan untuk menambahkan baris pada tabel MySQL, query untuk menambahkan data tersebut adalah sebagai berikut:  
INSERT INTO table\_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)

adapun contoh kode program PHP untuk menambahkan data tersebut adalah sebagai berikut:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){
        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";

    if (mysqli_query($conn, $sql)) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
?>
```

**Gambar 1.7** Memasukan Data

Jika kita melakukan perintah INSERT pada tabel yang menggunakan field dengan opsi AUTO\_INCREMENT, kita bisa mendapatkan ID dari baris yang terakhir diinputkan. Caranya menggunakan fungsi mysqli\_insert\_id(\$conn); nilai kembalian dari fungsi ini akan memberikan id dari record atau baris terakhir yang diinputkan.

Untuk menambahkan record dengan jumlah yang banyak, kita bisa menggunakan fungsi `mysqli_multi_query($conn, $sql)`, berikut adalah contoh program untuk menambahkan record dengan jumlah yang banyak:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Periksa Koneksi
if (!$conn){
```

**Gambar 1.8** Memasukan Data

```
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";

if (mysqli_multi_query($conn, $sql)) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

**Gambar 1.9** Memasukan Data

### Mengambil Data dari

Database Untuk mengambil data, statement SQL yang digunakan adalah `SELECT nama_kolom() FROM nama_tabel`, atau kita bisa menggunakan karakter `*` untuk memilih semua kolom yang ada pada tabel. Pengambilan data dengan kriteria tertentu bisa dilakukan dengan menggunakan

statement WHERE setelah nama\_tabel. Adapun contoh kode program untuk mengambil data dari database adalah sebagai berikut:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Memeriksa Koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}
```

**Gambar 1.10** Mengambil Data

```
$sql = "SELECT id, firstname, lastname FROM
MyGuests"; $result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Menampilkan data pada setiap baris
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

**Gambar 1.11** Mengambil Data

### Menghapus Data

dari Database Statement DELETE digunakan untuk menghapus baris data atau records dari tabel. Clause WHERE digunakan untuk menspesifikasikan baris yang akan dihapus. Jika statement DELETE digunakan tanpa menggunakan clausa WHERE, maka semua record yang ada pada tabel akan dihapus.



Jika kita memiliki data seperti tabel “MyGuest” seperti terlihat diatas, kemudian kita ingin menghapus data dengan nama depan Julie, maka kode program untuk menghapus data tersebut adalah sebagai berikut:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Membuat Koneksi
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

**Gambar 1.12 Menghapus**

```
// Memeriksa koneksi
if (!$conn){
    die("Connection failed: " . mysqli_connect_error());
}

// sql untuk menghapus record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

**Gambar 1.13 Menghapus**

### Update Data dalam Database

Untuk melakukan perubahan data di dalam database, statement UPDATE digunakan, yaitu sebagai berikut:

*UPDATE nama\_tabel*

*SET kolom1=nilai1, kolom2=nilai2, ...*

*WHERE kolom\_penentu=nilai\_penentu ...*

Jika statement ini digunakan untuk update data ke dua pada tabel “MyGuest” diatas untuk mengganti nama belakang dari “Moe” menjadi “Doe”, maka kode program untuk update tersebut adalah sebagai berikut:

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Membuat Koneksi
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Memeriksa Koneksi
    if (!$conn){
```

**Gambar 1.4** Update Data

```
        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

    if (mysqli_query($conn, $sql)) {
        echo "Record updated successfully";
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }

    mysqli_close($conn);
?>
```

**Gambar 1.5** Update Data

## BAB II

### PEMBAHASAN

#### Tugas Praktikum

1. Buatlah database pada MySql untuk menyimpan data-data pegawai beserta relasi tabelnya.
2. Buatlah program untuk menyimpan, menghapus dan mengubah data-data pegawai tersebut.

#### Database

Pada tabel department, ada dua buah kolom yang dibuat yaitu id yang bertipe data int dan nama\_depart yang bertipe data varchar. Untuk membuat tabel department diperlukan query sebagai berikut.

```
CREATE TABLE `department` (  
  `id` int(11) NOT NULL,  
  `nama_depart` varchar(20) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Setelah tabel dibuat, selanjutnya adalah menambahkan primary key, primary key berfungsi sebagai nilai utama yang nomor unik dari inputan data pada suatu tabel. Berikut query pembuatan primary key pada tabel department.

```
ALTER TABLE `department`  
  
ADD PRIMARY KEY (`id`);
```

Agar data terurut, maka diberikan auto increment sehingga data akan terurut sesuai inputan. Berikut query untuk auto increment.

```
ALTER TABLE `department`  
  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
```

Setelah tabel department selesai, tabel selanjutnya adalah tabel employee yang akan menyimpan data dari pegawai. Berikut query membuat tabel pegawai.

```
CREATE TABLE `employee` (  
  
  `id_employe` int(11) NOT NULL,  
  
  `nama` varchar(20) NOT NULL,  
  
  `id_department` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Untuk membuat primary key pada tabel employee digunakan query sebagai berikut.

```
ALTER TABLE `employee`  
  
ADD PRIMARY KEY (`id_employe`),  
  
ADD KEY `id_department` (`id_department`);
```

Sama seperti sebelumnya, auto increment juga digunakan pada tabel employee agar lebih dulu dalam mengurutkan data yang sudah diinput. Berikut querynya.

```
ALTER TABLE `employee`  
  
MODIFY `id_employe` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;
```

Selanjutnya adalah menambahkan foreign key pada tabel employee yang menyimpan nilai dari primary key tabel department. Untuk membuatnya digunakan query sebagai berikut.

```
ALTER TABLE `employee`  
  
    ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`id_department`) REFERENCES `department` (`id`),  
  
    ADD CONSTRAINT `employee_ibfk_2` FOREIGN KEY (`id_department`) REFERENCES `department` (`id`);  
  
COMMIT;
```

### Full Code Pembuatan Database

```
-- phpMyAdmin SQL Dump  
  
-- version 5.1.0  
  
-- https://www.phpmyadmin.net/  
  
--  
  
-- Host: 127.0.0.1  
  
-- Waktu pembuatan: 09 Apr 2021 pada 08.01  
  
-- Versi server: 10.4.18-MariaDB  
  
-- Versi PHP: 7.4.16  
  
  
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";  
  
START TRANSACTION;  
  
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--

-- Database: `bab3`

--

-----

--

-- Struktur dari tabel `department`

--

CREATE TABLE `department` (
  `id` int(11) NOT NULL,
  `nama_depart` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--  
  
-- Dumping data untuk tabel `department`  
  
--  
  
INSERT INTO `department` (`id`, `nama_depart`) VALUES  
  
(1, 'Telkom'),  
  
(2, 'Indofood'),  
  
(4, 'Sinar Mas');  
  
-----  
  
--  
  
-- Struktur dari tabel `employee`  
  
--  
  
CREATE TABLE `employee` (  
  
  `id_employe` int(11) NOT NULL,  
  
  `nama` varchar(20) NOT NULL,  
  
  `id_department` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--
```

```
-- Dumping data untuk tabel `employee`

--

INSERT INTO `employee` (`id_employe`, `nama`, `id_department`) VALUES
(1, 'Adi', 1),
(2, 'Yudi', 2),
(3, 'Dedi', 3);

--

-- Indexes for dumped tables

--

-- Indeks untuk tabel `department`

--

ALTER TABLE `department`

  ADD PRIMARY KEY (`id`);

--

-- Indeks untuk tabel `employee`

--

ALTER TABLE `employee`
```



```
ADD PRIMARY KEY (`id_employe`),

ADD KEY `id_department` (`id_department`);


--

-- AUTO_INCREMENT untuk tabel yang dibuang

--


--

-- AUTO_INCREMENT untuk tabel `department`

--

ALTER TABLE `department`

    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;


--

-- AUTO_INCREMENT untuk tabel `employee`

--

ALTER TABLE `employee`

    MODIFY `id_employe` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;


--

-- Ketidakeleluasaan untuk tabel pelimpahan (Dumped Tables)

--
```

```

--
-- Ketidakeleluasaan untuk tabel `employee`
--

ALTER TABLE `employee`

    ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`id_department`)
REFERENCES `department` (`id`),

    ADD CONSTRAINT `employee_ibfk_2` FOREIGN KEY (`id_department`)
REFERENCES `department` (`id`);

COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

## Department

Halaman department dibuat sebagai tempat untuk menampilkan inputan data dari tempat kerja employee. Halaman ini terhubung dengan fungsi melalui sintaks “require 'functiondepart.php';”.

Semua inputan atau kondisi terkini dari tabel department ditampilkan dengan query “\$departs = query("SELECT \* FROM department");”.

Agar user bisa menambahkan data pada tabel department, halaman ini dihubungkan menggunakan hyperlink dengan sintaks “<a href="department\_tambah.php">” yang mengarah ke halaman “department\_tambah.php”.

Untuk menghapus data dari tabel department, diberikan hyperlink yang mengarah ke “department\_delete.php”, berbeda dengan halaman tambah, untuk melakukan delete user tidak perlu berpindah halaman, user hanya perlu menekan pilihan hapus dan juga konfirmasinya. Berikut sintaksnya “<a href="department\_delete.php?id=<?php echo \$depart["id"]; ?>" onclick="return confirm('yakin?');">hapus</a>”.

Jika ingin berpindah ke halaman employee, juga disediakan button memiliki hyperlink yang akan mengarahkan user ke halaman tersebut hanya dengan mengkliknya, berikut adalah sintaks hyperlink ke halaman employee “<a href="employee.php">”.

### Full Code Tabel Department

```
<?php

    require 'functiondepart.php';

    $departs = query("SELECT * FROM department");

?>

<!doctype html>

<html lang="en">

    <head>

        <!-- Required meta tags -->

        <meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.c
ss" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<title>BAB3</title>

</head>

<body>

<div class="container">

  <div class="row">

    <div class="col mt-5">

      <a href="department_tambah.php">

        <div class="div btn btn-primary mb-4">

          Tambah Data

        </div>

      </a>

    </div>

  </div>

</div>
```

```
<a href="employee.php">

    <div class="div btn btn-primary mb-4">

        Kembali Ke Halaman Employe

    </div>

</a>

<table class="table table-striped">

    <thead>

        <tr>

            <th scope="col">No</th>

            <th scope="col">>Nama Department</th>

            <th scope="col">>Action</th>

        </tr>

    </thead>

    <tbody>

        <?php $i=1; ?>

        <?php foreach($departs as $depart) : ?>

            <td><?php echo $i; ?></td>

            <td><?php echo $depart["nama_depart"]; ?></td>

            <td>
```

```

                <a href="department_delete.php?id=<?php echo
$depart["id"]; ?>" onclick="return confirm('yakin?');">hapus</a>

            </td>

        </tr>

    </tbody>

    <?php $i++; ?>

    <?php endforeach; ?>

</table>

</div>

</div>

</div>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.m
in.js" integrity="sha384-

```

```
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>

    </body>

</html>
```

## Function Department

Semua fungsi tambah dan hapus pada tabel department, fungsi ini diminta oleh halaman halaman website sehingga bisa ditampilkan via website.

Fungsi tambah memiliki sintaks “insert into (nama\_tabel) values (‘.....’,’.....’). untuk fungsi tambah, user perlu memasukkan data yang diinginkan pada kurung setelah values sesuai urutan kolom pada tabel. Berikut sintaks untuk fungsi tambah.

```
function tambah($data){

    global $db;

    // ambil data dari tiap elemen dalam form

    $nama_depart = htmlspecialchars($data["nama_depart"]);

    $query = "INSERT INTO department

                VALUES

                ('', '$nama_depart')"
```

```
        ";

    // query insert data

    mysqli_query($db, $query);

    return mysqli_affected_rows($db);

}
```

Fungsi hapus memiliki sintaks “delete from (nama\_tabel) WHERE (nama\_kolom) = (nilai\_kolom)”. Untuk fungsi delete gunakan kolom id sebagai syarat yang harus dipenuhi agar fungsi hapus bisa berjalan. Berikut sintaks untuk fungsi hapus.

```
function hapus($id){

    global $db;

    mysqli_query($db, "DELETE FROM department WHERE id = $id");

    return mysqli_affected_rows($db);

}
```

### **Full Code Function Department**

```
<?php

    $db = mysqli_connect("localhost", "root", "", "bab3");
```



```
function query($query){  
  
    global $db;  
  
    $result = mysqli_query($db, $query);  
  
    $rows = [];  
  
    while($row = mysqli_fetch_assoc($result)){  
  
        $rows[]=$row;  
  
    }  
  
    return $rows;  
  
}
```

```
function tambah($data){  
  
    global $db;  
  
    // ambil data dari tiap elemen dalam form  
  
    $nama_depart = htmlspecialchars($data["nama_depart"]);  
  
  
  
    $query = "INSERT INTO department  
  
                VALUES  
  
                ('', '$nama_depart')  
  
                ";  
  
    // query insert data
```

```
mysqli_query($db, $query);

return mysqli_affected_rows($db);
}

function hapus($id){
    global $db;

    mysqli_query($db, "DELETE FROM department WHERE id = $id");

    return mysqli_affected_rows($db);
}

?>
```

### **Tabel Employee**

Halaman employee dibuat sebagai tempat untuk menampilkan inputan data dari tempat kerja employee. Halaman ini terhubung dengan fungsi melalui sintaks “require 'functionemployee.php';”.

Semua inputan atau kondisi terkini dari tabel employee ditampilkan dengan query “\$employs = query("SELECT \* FROM employee INNER JOIN department ON employee.id\_department=department.id");”.

Terdapat INNER JOIN yang menggabungkan tabel employee dan tabel department dengan nilai kolom yang sama yaitu id\_department pada tabel employee dengan kolom id pada tabel department.

Agar user bisa menambahkan data pada tabel employee, terdapat button tambah data di halaman ini yang dihubungkan menggunakan hyperlink dengan sintaks “<a href="employee\_tambah.php">” yang mengarahkan user ke halaman “employee\_tambah.php”.

Untuk mengubah data, halaman ini mengakses fungsi ubah data yang ada pada functionemploye.php. dengan mengklik action ubah, user akan diarahkan ke halaman Ubah Data Employee dengan adanya sintaks berikut “<a href="employee\_ubah.php?id=<?php echo \$employe["id\_employe"]; ?>">ubah</a>”.

Untuk menghapus data dari tabel department, diberikan hyperlink yang mengarah ke “employee\_delete.php”, berbeda dengan halaman tambah, untuk melakukan delete user tidak perlu berpindah halaman, user hanya perlu menekan pilihan action hapus dan juga konfirmasinya. Berikut sintaksnya “<a href="employee\_delete.php?id=<?php echo \$depart["id"]; ?>" onclick="return confirm('yakin?');">hapus</a>”.

Jika ingin berpindah ke halaman employee, juga disediakan button memiliki hyperlink yang akan mengarahkan user ke halaman tersebut hanya dengan mengkliknya, berikut adalah sintaks hyperlink ke halaman department “<a href="department.php">”.

### Full Code Tabel Employee

```
<?php

require 'functionemploye.php';

$employs = query("SELECT * FROM employee INNER JOIN department ON
employee.id_department=department.id");
```

```
?>

<!doctype html>

<html lang="en">

  <head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <!-- Bootstrap CSS -->

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.c
ss" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <title>BAB3</title>

  </head>

  <body>

    <div class="container">

      <div class="row">

        <div class="col mt-5">
```

```
<a href="employe_tambah.php">

    <div class="div btn btn-primary mb-4">

        Tambah Data

    </div>

</a>
```

```
<a href="department.php">

    <div class="div btn btn-primary mb-4">

        Ke Halaman Department

    </div>

</a>
```

```
<table class="table table-striped">

    <thead>

        <tr>

            <th scope="col">No</th>

            <th scope="col">Nama</th>

            <th scope="col">Dependent</th>

            <th scope="col">Action</th>

        </tr>
```

```
</thead>

<tbody>

<?php $i=1; ?>

<?php foreach($employs as $employee) : ?>

    <td><?php echo $i; ?></td>

    <td><?= $employee["nama"]; ?></td>

    <td><?= $employee["nama_depart"]; ?></td>

    <td>

        <a href="employee_ubah.php?id=<?php echo
$employee["id_employee"]; ?>">ubah</a> |

        <a href="employee_delete.php?id=<?php echo
$employee["id_employee"]; ?>" onclick="return confirm('yakin?');">hapus</a>

    </td>

</tr>

</tbody>

<?php $i++; ?>

<?php endforeach; ?>

</table>

</div>

</div>

</div>
```

```

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.m
in.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>

</body>

</html>

```

## Function Employee

Semua fungsi pada tabel employee, fungsi ini diminta oleh halaman halaman website sehingga bisa ditampilkan via website.

Fungsi tambah memiliki sintaks “insert into (nama\_tabel) values (‘.....’,’.....’). untuk fungsi tambah, user perlu memasukkan data yang diinginkan pada kurung setelah values sesuai urutan kolom pada tabel. Berikut sintaks untuk fungsi tambah.

```

function tambah($data){

    global $db;

    // ambil data dari tiap elemen dalam form

    $nama = htmlspecialchars($data["nama"]);

    $id_department = htmlspecialchars($data["id_department"]);

    $query = "INSERT INTO employee

                VALUES

                ('', '$nama', '$id_department')

                ";

    // query insert data

    mysqli_query($db, $query);

    return mysqli_affected_rows($db);

}

```

Fungsi ubah digunakan untuk mengubah inputan yang ada, sintaks untuk fungsi ubah adalah “update (nama\_tabel) set (nama\_kolom) = (nilai\_baru) where (nama\_kolom) = (nilai\_kolom);”.

Sintaks “set” digunakan untuk mengatur data baru pada kolom yang diinginkan dengan “where sebuah kolom” sebagai kondisi yang harus dipenuhi. Berikut sintaks untuk fungsi ubah.

```

function ubah($data){

    global $db;

```



```

$id_employe = $data["id_employe"];

// ambil data dari tiap elemen dalam form

$nama = htmlspecialchars($data["nama"]);

$id_department = htmlspecialchars($data["id_department"]);

$query = "UPDATE employee SET

            nama = '$nama',

            id_department = '$id_department'

            WHERE id_employe = $id_employe

        ";

// query insert data

mysqli_query($db, $query);

return mysqli_affected_rows($db);

}

```

Fungsi hapus memiliki sintaks “delete from (nama\_tabel) WHERE (nama\_kolom) = (nilai\_kolom)”. Untuk fungsi delete gunakan kolom id\_employee sebagai syarat yang harus dipenuhi agar fungsi hapus bisa berjalan. Masukkan nilai id\_employee yang ingin dihapus. Berikut sintaks untuk fungsi hapus.

```

function hapus($id){

    global $db;

    mysqli_query($db, "DELETE FROM employee WHERE id_employe = $id");

    return mysqli_affected_rows($db);
}

```

```
}
```

### Full Code Function Employee

```
<?php

$db = mysqli_connect("localhost", "root", "", "bab3");

function query($query){

    global $db;

    $result = mysqli_query($db, $query);

    $rows = [];

    while($row = mysqli_fetch_assoc($result)){

        $rows[]=$row;

    }

    return $rows;

}

function tambah($data){

    global $db;

    // ambil data dari tiap elemen dalam form

    $nama = htmlspecialchars($data["nama"]);
```

```

$id_department = htmlspecialchars($data["id_department"]);

$query = "INSERT INTO employee
        VALUES
        ('', '$nama', '$id_department')
        ";

// query insert data
mysqli_query($db, $query);

return mysqli_affected_rows($db);
}

function ubah($data){
    global $db;

    $id_employe = $data["id_employe"];

    // ambil data dari tiap elemen dalam form
    $nama = htmlspecialchars($data["nama"]);

    $id_department = htmlspecialchars($data["id_department"]);

    $query = "UPDATE employee SET
            nama = '$nama',
            id_department = '$id_department'
            WHERE id_employe = $id_employe
    ";

```

```
        ";

        // query insert data

        mysqli_query($db, $query);

        return mysqli_affected_rows($db);
    }

    function hapus($id){

        global $db;

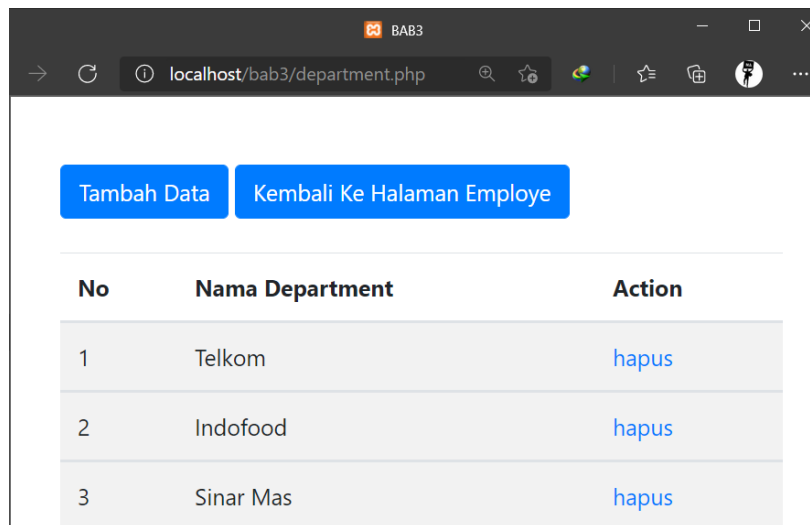
        mysqli_query($db, "DELETE FROM employee WHERE id_employe = $id");

        return mysqli_affected_rows($db);

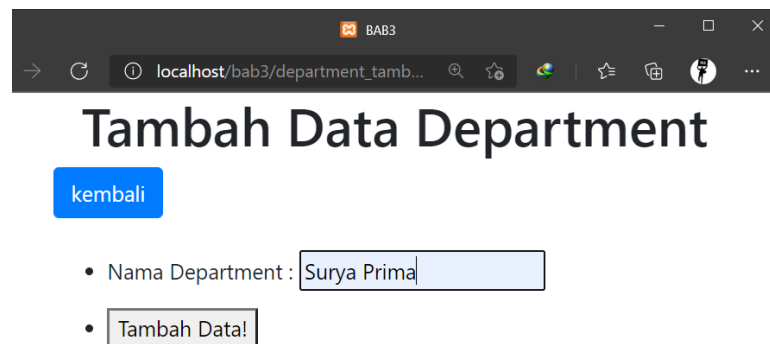
    }

?>
```

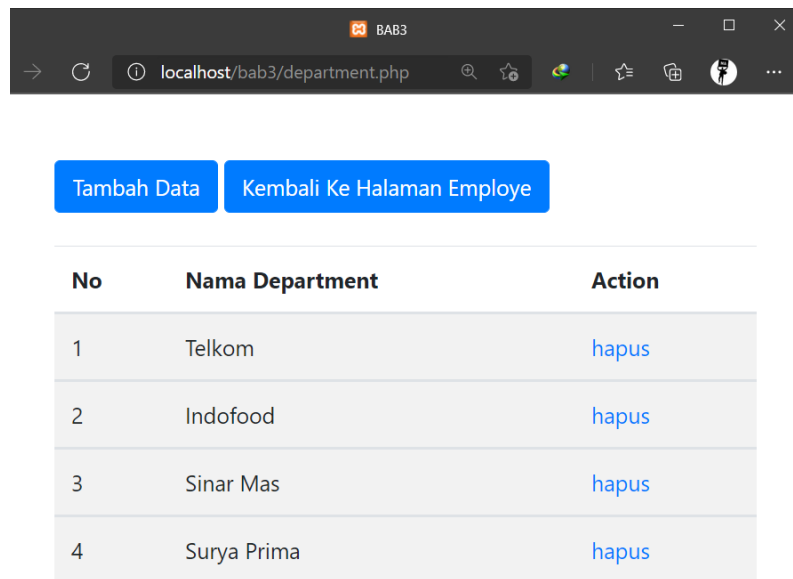
Berikut adalah output program di atas.



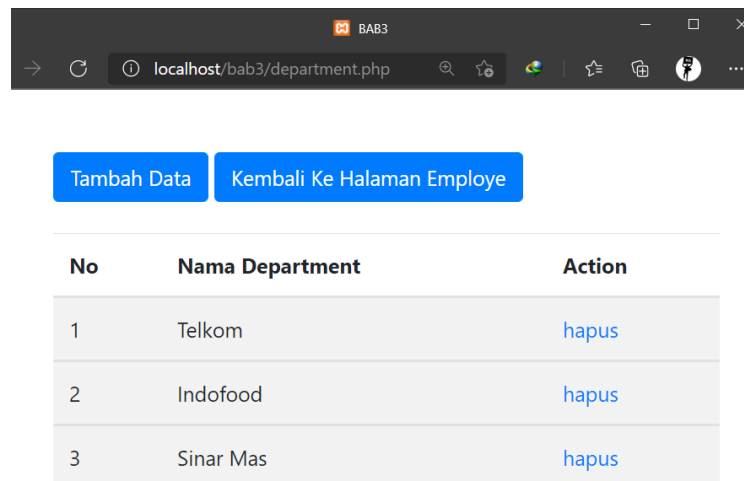
**Gambar 2.1** Tabel department.



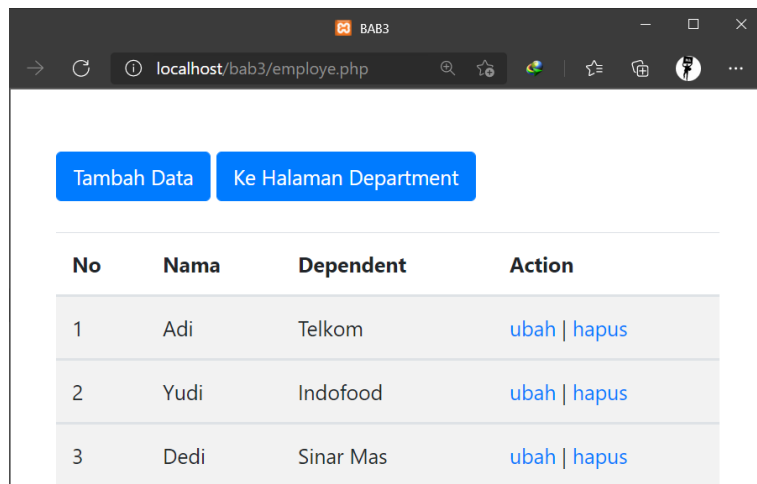
**Gambar 2.2** Tambah department,



**Gambar 2.3** Hasil tambah department.

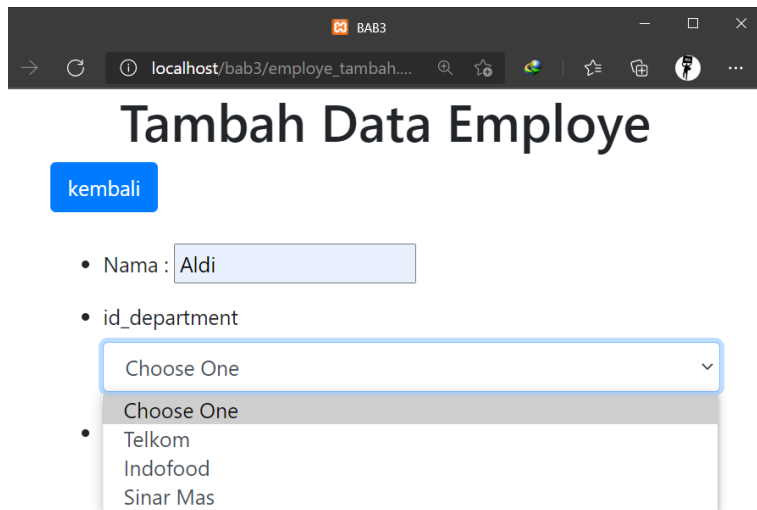


**Gambar 2.4** Hasil hapus department.



No	Nama	Dependent	Action
1	Adi	Telkom	<a href="#">ubah</a>   <a href="#">hapus</a>
2	Yudi	Indofood	<a href="#">ubah</a>   <a href="#">hapus</a>
3	Dedi	Sinar Mas	<a href="#">ubah</a>   <a href="#">hapus</a>

**Gambar 2.5** Tabel employee.



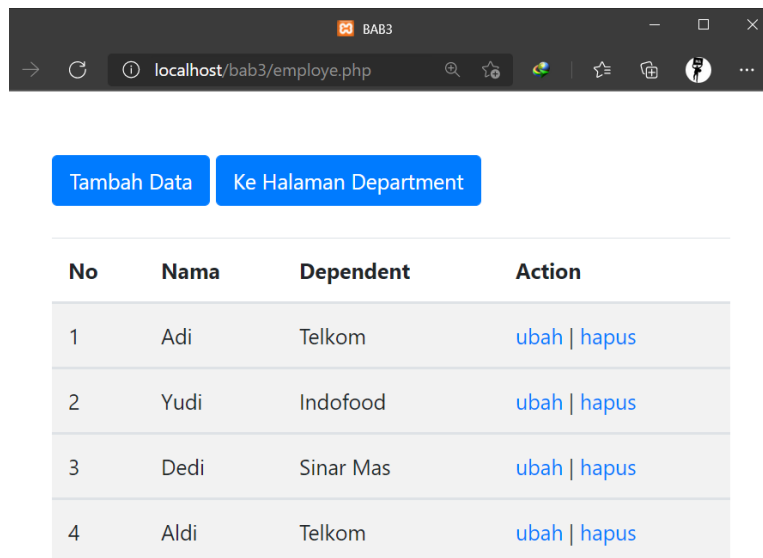
## Tambah Data Employe

[kembali](#)

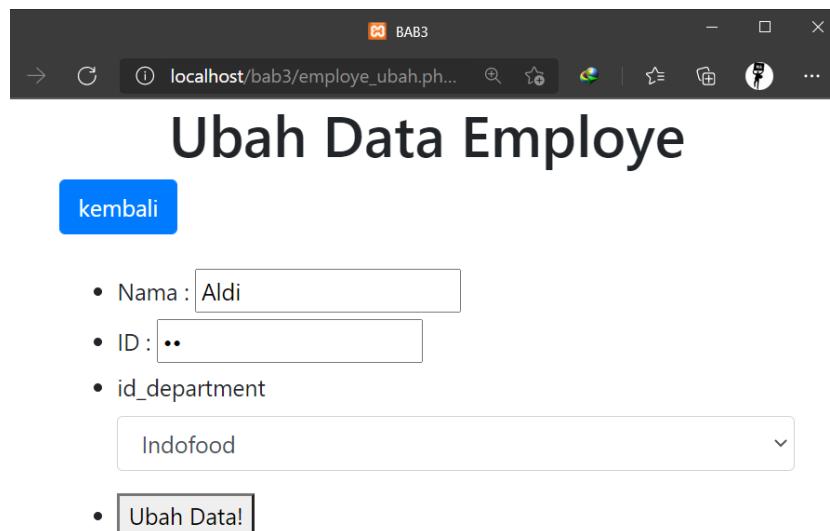
- Nama :
- id\_department
 

- Choose One
  - Telkom
  - Indofood
  - Sinar Mas

**Gambar 2.6** Tambah employee.

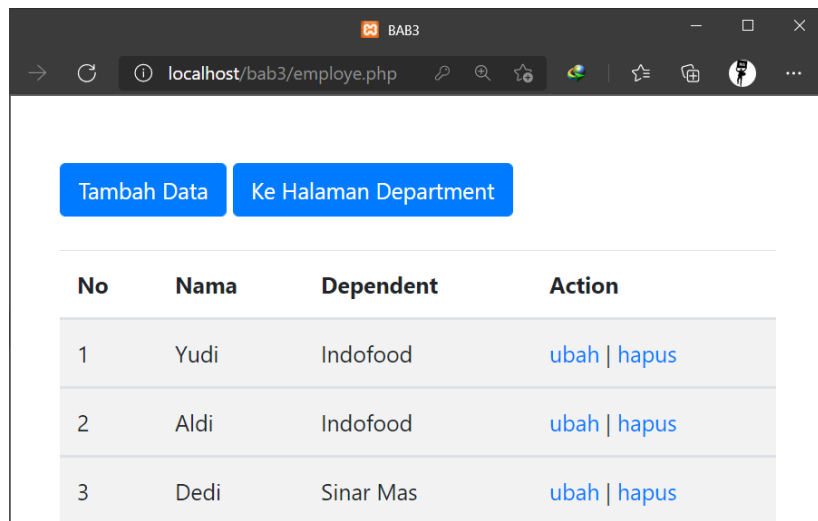


**Gambar 2.7** Hasil tambah employee.

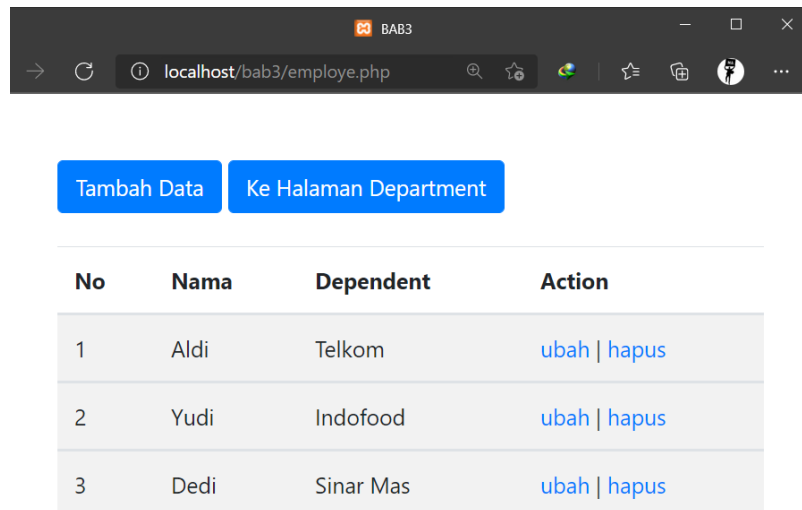


**Gambar 2.8** Ubah employee.





**Gambar 2.9** Hasil ubah employee.



**Gambar 2.10** Hasil hapus employee.



### **BAB III**

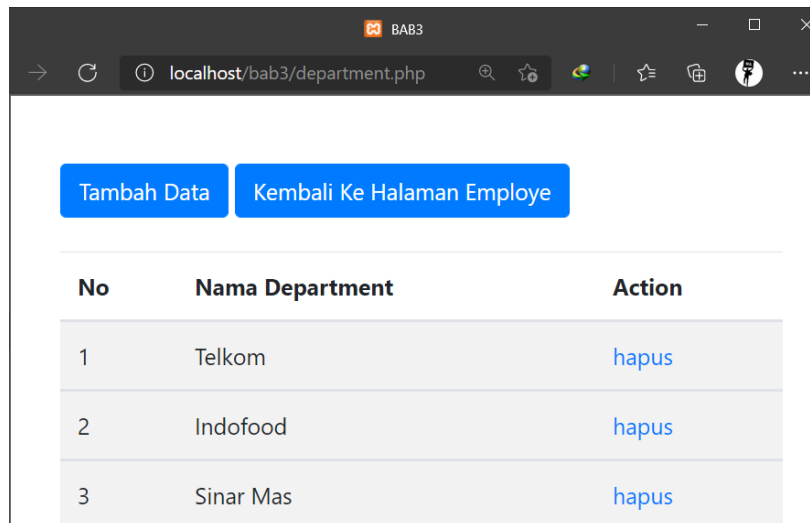
#### **KESIMPULAN**

Penggunaan mysql digunakan untuk menangani database, user bisa membuat tabel untuk mengatur data-data yang sudah diinputkan. Penanganan database bisa dilakukan melalui aplikasi seperti SQLyog maupun localhost. Agar bisa diakses melalui website, digunakan php untuk membuat halaman dan menghubungkan website dengan database mysql.

## **DAFTAR PUSTAKA**

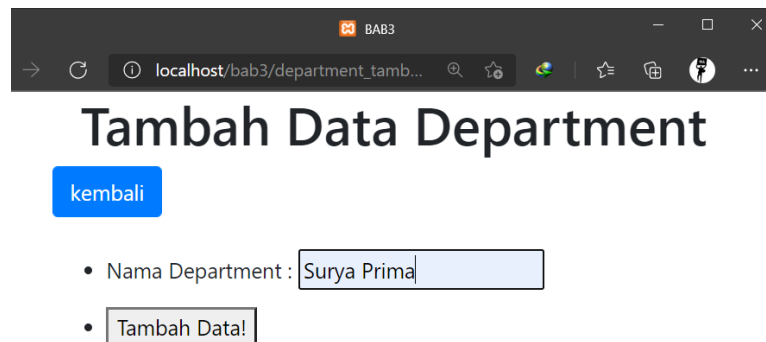
Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.*

## LAMPIRAN



No	Nama Department	Action
1	Telkom	<a href="#">hapus</a>
2	Indofood	<a href="#">hapus</a>
3	Sinar Mas	<a href="#">hapus</a>

**Gambar 2.1** Tabel department.

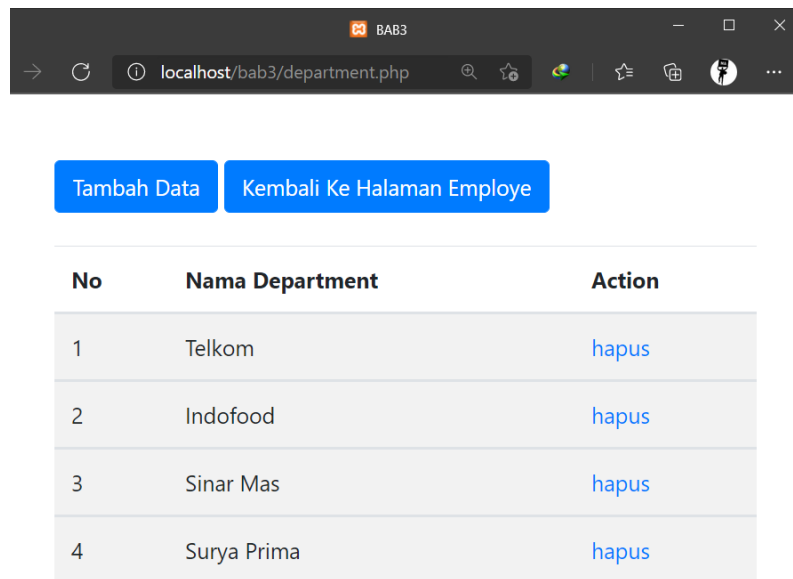


### Tambah Data Department

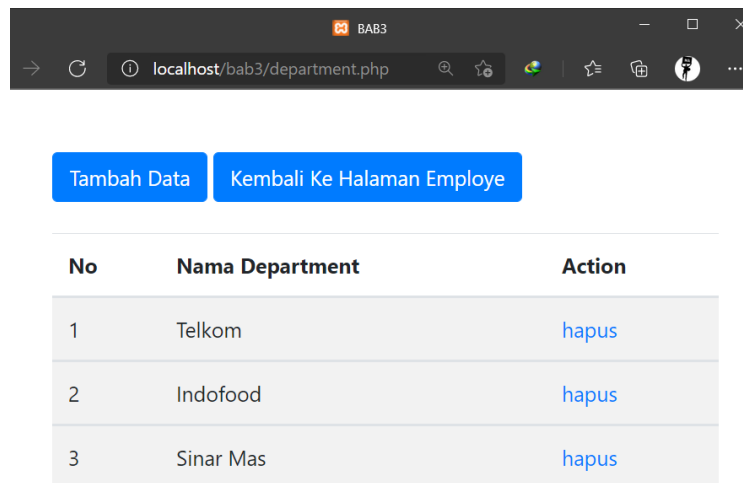
[kembali](#)

- Nama Department :
- 

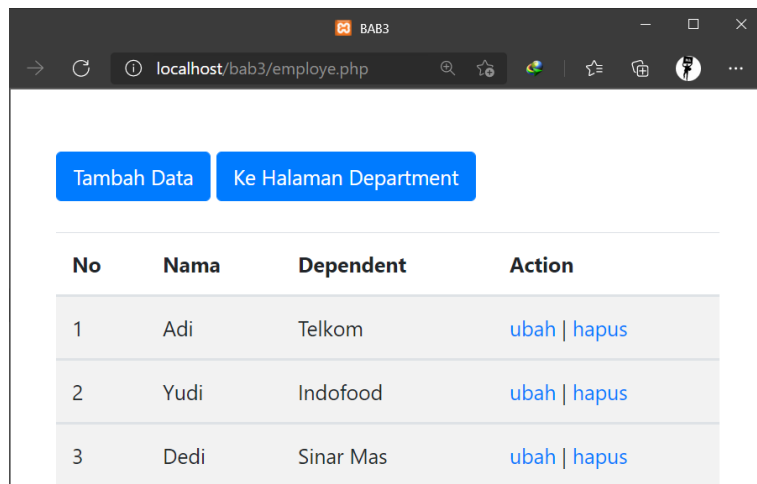
**Gambar 2.2** Tambah department,



**Gambar 2.3** Hasil tambah department.

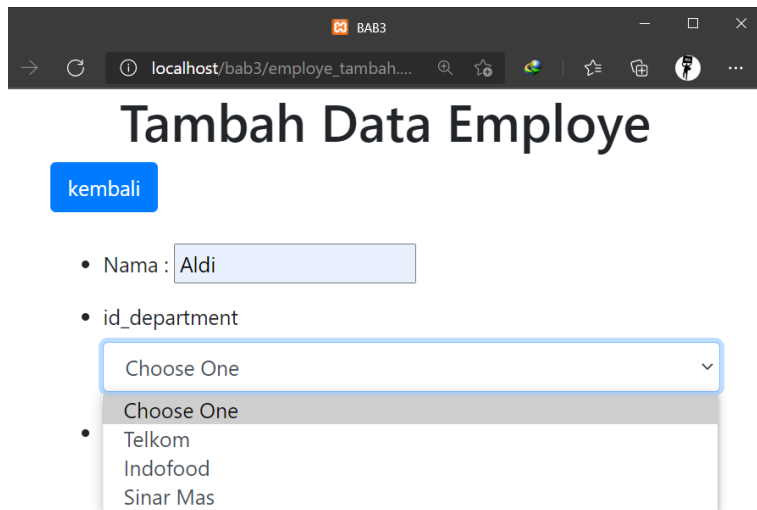


**Gambar 2.4** Hasil hapus department.



No	Nama	Dependent	Action
1	Adi	Telkom	<a href="#">ubah</a>   <a href="#">hapus</a>
2	Yudi	Indofood	<a href="#">ubah</a>   <a href="#">hapus</a>
3	Dedi	Sinar Mas	<a href="#">ubah</a>   <a href="#">hapus</a>

**Gambar 2.5** Tabel employee.



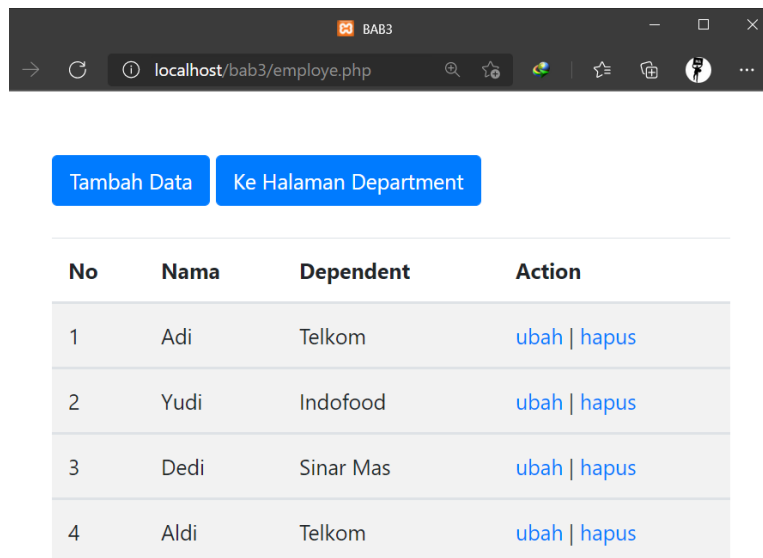
## Tambah Data Employee

[kembali](#)

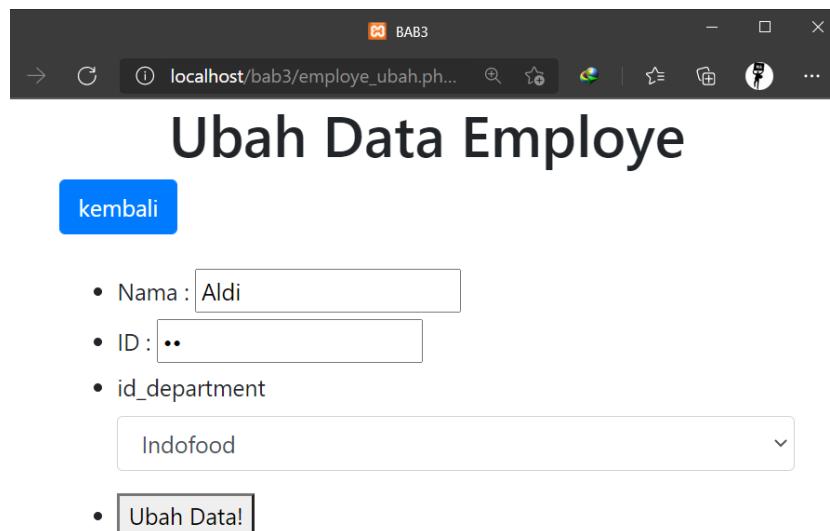
- Nama :
- id\_department
 

- Choose One
  - Telkom
  - Indofood
  - Sinar Mas

**Gambar 2.6** Tambah employee.

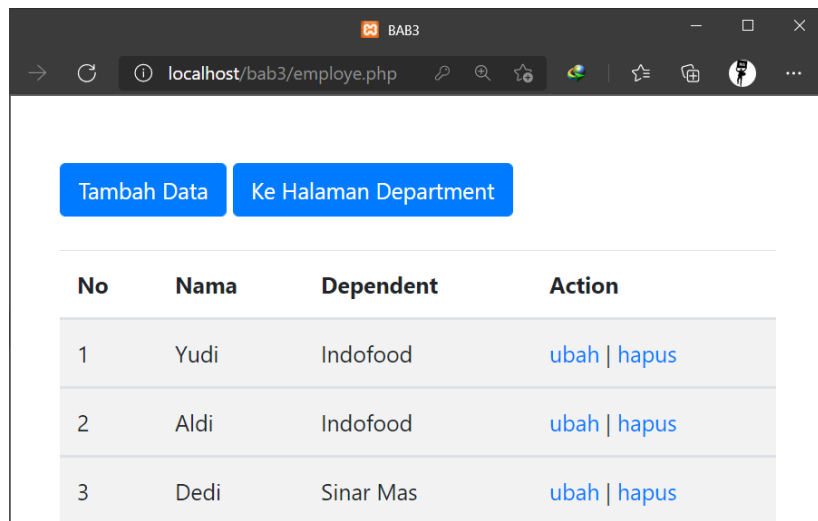


**Gambar 2.7** Hasil tambah employee.

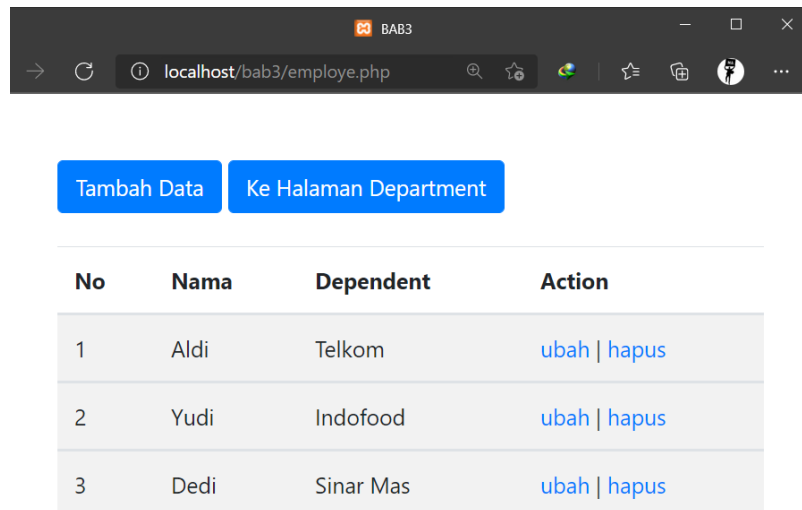


**Gambar 2.8** Ubah employee.





**Gambar 2.9** Hasil ubah employee.



**Gambar 2.10** Hasil hapus employee.