



## Universitas Kristen Duta Wacana

### Fakultas Teknologi Informasi

#### Prodi Informatika

Jln. Dr. Wahidin Sudiro Husodo No 5-25

Daerah Istimewa Yogyakarta 55224

<b>Mata Kuliah</b>	: Pemrograman Website
<b>Kode Mata Kuliah</b>	: TI0243
<b>Periode</b>	: Semester Genap 2020/2021
<b>Dosen Pengampu</b>	: Danny Sebastian S.Kom., M.M., M.T. : Maria Nila Anggia Rini S.T., M.Tl. : Agata Filiana S.Kom., M.Cs.

## BAB 7 PHP DATABASE

### A. CPL Pertemuan

- Memperkenalkan syntax-syntax pada Bahasa pemrograman PHP
- Mengakses database MariaDB atau MySQL menggunakan Bahasa pemrograman PHP

### B. Alat & Bahan

Kegiatan praktikum pada bagian ini membutuhkan perangkat keras (*hardware*) atau komputer dengan spesifikasi minimum:

- Processor : Intel Core i3
- Memory : 2 GB
- Hard disk : kapasitas kosong penyimpanan sebesar 3 GB

Selain perangkat keras, kegiatan praktikum bagian ini juga membutuhkan perangkat lunak (*software*):

- Sistem Operasi (Windows, Linux, atau MacOSX)
- Text editor (Atom, Visual Studio Code, atau editor lain)
- Browser (Google Chrome, Opera, atau Mozilla Firefox)
- Xampp (Web Server Apache & Database MariaDB)

### C. HTML Form & Input Type

#### C.1. Form

HTML Form atau `<form>` adalah tag/elemen HTML untuk menerima input dari pengguna pada halaman website. Data input yang dikirim adalah tag input yang berada di dalam/merupakan anak dari

tag `<form>`. Data input disesuaikan dengan input type HTML yang ada. Tag `<form>` memiliki beberapa attribute yang perlu diperhatikan, yaitu:

- **method** : merupakan penentu bagaimana input tersebut akan diproses, biasanya menggunakan POST atau GET method. Ada method lain, tetapi tidak kita bahas untuk praktikum ini.
- **action** : merupakan file-path dari file yang akan mengelola input dari pengguna.
- **Enctype** : merupakan penentu bagaimana form-data diolah di server. Digunakan ketika hendak melakukan upload file / menerima input berupa file.

### C.2. Text

HTML input `type="text"` digunakan untuk menerima input pengguna berupa text yang bersifat 1 baris. Contoh penggunaan untuk field nama-depan, nama-belakang, nama-ibu-kandung, dll. Attribute khusus untuk input type text adalah:

- **maxlength** : mendefinisikan Panjang maksimum dari text yang dapat diterima
- **minlength** : mendefinisikan Panjang minimum dari text yang dapat diterima
- **required** : mendefinisikan data harus diisi pada saat disimpan
- **placeholder** : mendefinisikan label/keterangan field input.
- **value** : mendefinisikan nilai dari input field

### C.3. Checkbox

HTML input `type="checkbox"` digunakan untuk menerima inputan yang bersifat true/false dan pengguna dapat memilih lebih dari satu opsi. Contoh penggunaan untuk field hobi, keahlian, dll. Attribute khusus untuk input type checkbox adalah:

- **checked** : mendefinisikan bahwa input field tersebut bernilai true, dan value-nya akan dikirim ke server untuk diproses.
- **value** : mendefinisikan nilai dari input field

### C.4. Date

HTML input `type="date"` digunakan untuk menerima inputan berupa tanggal menggunakan datepicker. Contoh penggunaan untuk tanggal-lahir, tanggal-mulai kerja, dll. Attribute khusus untuk input type date adalah:

- **min** : mendefinisikan tanggal minimum yang boleh dipilih
- **max** : mendefinisikan tanggal maksimum yang boleh dipilih
- **value** : mendefinisikan nilai dari input field

### C.5. File

HTML input `type="file"` digunakan untuk menerima inputan berupa file data. Contoh penggunaan untuk upload gambar, upload pdf, dll. Attribute khusus untuk input type file adalah:

- `accept` : mendefinisikan file type yang diperbolehkan
- `value` : mendefinisikan nilai dari input field

#### C.6. Email

HTML input `type="email"` digunakan untuk menerima input pengguna berupa text dengan format email. Attribute khusus untuk input type email adalah:

- `maxlength` : mendefinisikan Panjang maksimum dari text yang dapat diterima
- `minlength` : mendefinisikan Panjang minimum dari text yang dapat diterima
- `required` : mendefinisikan data harus diisi pada saat disimpan
- `placeholder` : mendefinisikan label/keterangan field input.
- `Pattern` : mendefinisikan regular expression untuk validasi pattern alamat email
- `value` : mendefinisikan nilai dari input field

#### C.7. Number

HTML input `type="number"` digunakan untuk menerima input pengguna berupa text dengan format number atau angka. Attribute khusus untuk input type number adalah:

- `max` : mendefinisikan maksimum dari text yang dapat diterima
- `min` : mendefinisikan minimum dari text yang dapat diterima
- `required` : mendefinisikan data harus diisi pada saat disimpan
- `placeholder` : mendefinisikan label/keterangan field input.
- `value` : mendefinisikan nilai dari input field

#### C.8. Password

HTML input `type="password"` digunakan untuk menerima input pengguna berupa text. Yang membedakan dengan input type text adalah ditampilkan menggunakan bintang/disensor. Attribute khusus untuk input type password adalah:

- `max` : mendefinisikan Panjang maksimum dari text yang dapat diterima
- `min` : mendefinisikan Panjang minimum dari text yang dapat diterima
- `required` : mendefinisikan data harus diisi pada saat disimpan
- `placeholder` : mendefinisikan label/keterangan field input.
- `value` : mendefinisikan nilai dari input field

#### C.9. Radio

HTML input `type="radio"` digunakan untuk menerima inputan yang bersifat true/false dan pengguna hanya bisa memilih 1 opsi. Contoh penggunaan untuk field jenis kelamin, dll. Attribute khusus untuk input type radio adalah:

- `checked` : mendefinisikan bahwa input field tersebut bernilai true, dan value-nya akan dikirim ke server untuk diproses.
- `value` : mendefinisikan nilai dari input field

### C.10. Hidden

HTML input `type="hidden"` digunakan untuk memberikan parameter yang tersembunyi dari pengguna. Attribute khusus untuk input type number adalah:

- `value` : mendefinisikan nilai dari input field

### C.11. Button & Reset & Submit

HTML input `type="button"` `type="reset"` `type="submit"` digunakan untuk tombol. Yang membedakan ketiganya adalah, input type button bersifat multi-purpose, atau digunakan untuk segala keperluan, sedangkan input type reset digunakan untuk mengosongkan input field lain dalam 1 form yang sama, sedangkan input type submit digunakan untuk menjalankan form ke attribute action milik tag `<form>`.

## D. PHP Database

Hal mendasar untuk mengelola konten pada sebuah halaman website adalah SELECT, INSERT, UPDATE, dan DELETE. Pada praktikum ini, akan dijelaskan bagaimana mengolah data dari pengguna. Topik yang digunakan untuk penjelasan bagian ini adalah entitas Mahasiswa.

### D.1. Database

Pada pengembangan website menggunakan PHP dan MariaDB/MySQL pada umumnya akan menggunakan xampp bundling. Didalam xampp bundling terdapat phpMyAdmin sebagai website untuk mengelola data pada database. Untuk mengakses phpMyAdmin dapat melalui URL <http://localhost/phpmyadmin>.

Untuk melanjutkan praktikum, buatlah database php2, dengan tabel mahasiswa. Kolom dari tabel mahasiswa dapat dilihat pada gambar dibawah ini.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 nim	varchar(8)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 nama	varchar(50)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 jenis_kelamin	varchar(1)	latin1_swedish_ci		No	None			Change  Drop  More

### D.2. Select Data

Select data, atau yang sering disebut dengan READ data biasanya merupakan tahap paling awal dalam pembuatan sebuah website. Pada bagian ini kita akan mencoba membuat sebuah tabel yang menampilkan data dari entitas mahasiswa. Logika berpikir untuk tahap ini:

1. Buat koneksi ke database

2. Ambil daftar mahasiswa dari database menggunakan query SELECT
3. Periksa apakah ada daftar mahasiswa
4. Apabila ada, buat tabel header
5. Buat perulangan untuk menampilkan setiap baris

Langkah Kerja:

1. Buat sebuah halaman index.php, pada halaman ini, daftar mahasiswa akan ditampilkan pada sebuah tabel.
2. Pada bagian atas halaman, buat koneksi ke database.

```
// open connection
$servername = "127.0.0.1";
$username = "progweb";
$password = "progweb";
$databasename = "php2";
$conn = mysqli_connect($servername, $username, $password, $databasename) or die("Koneksi gagal.");
```

3. Ambil data mahasiswa menggunakan query SELECT.

```
//SELECT
$sql = "SELECT * FROM mahasiswa";
$result = mysqli_query($conn, $sql); //untuk run query
```

4. Periksa apakah ada daftar mahasiswa.

```
if(mysqli_num_rows($result) > 0) {
} else {
    echo "Tabel mahasiswa kosong.";
}
```

5. Buat header untuk tabel daftar mahasiswa. Header yang ditampilkan adalah NIM, nama, jenis kelamin, dan sebuah kolom tambahan untuk edit dan delete.

```

if(mysqli_num_rows($result) > 0) {
    ?>
    <table border=1>
        <thead>
            <th>NIM</th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
            <th>Action</th>
        </thead>
    </table>
    <?php
} else {
    echo "Tabel mahasiswa kosong.";
}
?>

```

6. Buat sebuah perulangan untuk menampilkan setiap baris data menggunakan tag `<tr>` dan `<td>`.

```

<?php
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>".$row["nim"]."</td>";
    echo "<td>".$row["nama"]."</td>";
    echo "<td>".$row["jenis_kelamin"]."</td>";
    echo "<td></td>";
    echo "</tr>";
}
?>

```

7. Hasil yang didapat adalah

NIM	Nama	Jenis Kelamin	Action
71150001	Danny Sebastian	L	
71150002	Maria Nila A	P	

### D.3. Insert Into

Insert into, atau yang sering disebut dengan CREATE data digunakan untuk menambah data mahasiswa. Pada bagian ini kita akan mencoba membuat sebuah form yang digunakan untuk menerima data mahasiswa baru. Logika berpikir untuk tahap ini:

1. Tampilkan form input
2. Pada saat disubmit, Periksa inputan sudah sesuai atau belum.

3. Bila input sudah sesuai, buat koneksi ke database, kemudian isikan data menggunakan query INSERT
4. Berikan notifikasi input sudah berhasil disimpan
5. Bila pada langkah 2 masih ada data yang tidak sesuai, berikan notifikasi kesalahan

Langkah Kerja:

1. Buat halaman form-mhs.php, halaman ini akan menampilkan sebuah form untuk melakukan edit dan insert data mahasiswa.
2. Isikan form dengan input type masing-masing field, berikan parameter action ke file yang akan mengelola input tersebut. Pada kasus ini file yang mengelola input adalah form-mhs.php atau dirinya sendiri.

```
<form action="form-mhs.php" method="POST">
  nim : <input type="number" name="nim" max="8"> <br>
  nama : <input type="text" name="nama" max="50"> <br>
  kelamin : <input type="radio" value="L" name="jenis_kelamin">Laki-laki
  <input type="radio" value="P" name="jenis_kelamin">Perempuan <br>
  <input type="submit">
</form>
```

3. Karena `action="form-mhs.php"` atau dirinya sendiri, maka pada bagian atas, sebelum tag `<html>`, tambahkan open connection.

```
// open connection
$servername = "127.0.0.1";
$username = "progweb";
$password = "progweb";
$databasename = "php2";
$conn = mysqli_connect($servername, $username, $password, $databasename) or die("Koneksi gagal.");
```

4. Selanjutnya, kita melakukan pengecekan, apakah ada data pada variable `$_POST`. Bila ada, maka lakukan proses insert ke database.

```
if($_POST){
  $nim = $_POST["nomer"];
  $nama = $_POST["nama"];
  $kelamin = $_POST["jenis_kelamin"];
  $sql = "INSERT INTO mahasiswa (nim, nama, jenis_kelamin) VALUES ('.$nim.', '.$nama.', '.$kelamin.')";
  if(mysqli_query($conn, $sql)) {
    echo "Berhasil mengisi data.";
  } else {
    echo "Gagal mengisi data.";
  }
}
```

5. Lakukan close connection



```
// close connection
mysqli_close($conn);
```

#### D.4. Update Data

Update data, atau yang sering disebut dengan UPDATE data digunakan untuk mengubah data mahasiswa. Pada bagian ini kita akan menggunakan form mahasiswa yang sudah dibuat pada tahap sebelumnya. Logika berpikir untuk tahap ini:

6. Pada halaman SELECT atau daftar mahasiswa, kolom action, tambahkan sebuah tombol/link menuju ke halaman form dengan membawa parameter GET id dari data yang hendak di edit.
7. Pada form mahasiswa, apabila ada data GET id, maka tambahkan sebuah `<input type="hidden" value="id">`
8. Pada saat disubmit, Periksa inputan sudah sesuai atau belum.
9. Bila input sudah sesuai, buat koneksi ke database, kemudian isikan data menggunakan query UPDATE dengan WHERE adalah id dari input type hidden
10. Berikan notifikasi input sudah berhasil diubah
11. Bila pada langkah 3 masih ada data yang tidak sesuai, berikan notifikasi kesalahan

Langkah Kerja:

1. Buka file index.php atau file untuk daftar mahasiswa.
2. Pada bagian perulangan data, kolom terakhir, tambahkan link untuk ke halaman form-mhs.php. Yang perlu diperhatikan adalah, id data GET yang dikirimkan.

```
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>".$row["nim"]."</td>";
    echo "<td>".$row["nama"]."</td>";
    echo "<td>".$row["jenis_kelamin"]."</td>";
    echo "<td>";
        echo "<a href='form-mhs.php?id=".$row['id']."'>Edit</a>";
    echo "</td>";
    echo "</tr>";
}
```

3. Buat sebuah percabangan, bila ada data GET, maka proses pengambilan data awal/data yang hendak dirubah. Lakukan pencarian ke database untuk mahasiswa yang memiliki id GET.



```

if($_GET){
    $id = $_GET['id'];
    $sql = "SELECT * FROM mahasiswa WHERE id='".$_GET['id']."'";
    $result = mysqli_query($conn, $sql);
    if(mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_assoc($result);
        $oldNim = $row["nim"];
        $oldNama = $row["nama"];
        $oldKelamin = $row["jenis_kelamin"];
    }
    else {
        echo "Data yang hendak diedit tidak ada.";
    }
}

```

4. Pada bagian `<form>`, tambahkan input hidden untuk id, dan isikan data awal berdasarkan data dari database.

```

<form action="form-mhs.php" method="POST">
    <?php
        if($id != null) {
            ?>
            <input type="hidden" name="id" value="<?php echo $id; ?>">
            <?php
        }
    ?>
    nim : <input type="number" name="nim" max="8" value="<?php echo $oldNim; ?>"> <br>
    nama : <input type="text" name="nama" max="50" value="<?php echo $oldNama; ?>"> <br>
    kelamin : <input type="radio" value="L" name="jenis_kelamin" <?php echo ($oldKelamin=='L')?'checked':''; ?> >Laki-laki
    <input type="radio" value="P" name="jenis_kelamin" <?php echo ($oldKelamin=='P')?'checked':''; ?> >Perempuan <br>
    <input type="submit">
</form>

```

5. Pada bagian INSERT data, buat percabangan untuk UPDATE data.

```

if($_POST){
    if($_POST['id'] != null){
        //update
    } else {
        //insert
        $nim = $_POST["nim"];
        $nama = $_POST["nama"];
        $kelamin = $_POST["jenis_kelamin"];
        $sql = "INSERT INTO mahasiswa (nim, nama, jenis_kelamin) VALUES ('".$_nim."', '".$_nama."', '".$_kelamin."')";
        if(mysqli_query($conn, $sql)) {
            echo "Berhasil mengisi data.";
        } else {
            echo "Gagal mengisi data.";
        }
    }
}

```

6. Tambahkan logika untuk melakukan UPDATE ke database.

```
if($_POST['id'] != null){
    //update
    $nim = $_POST["id"];
    $nim = $_POST["nim"];
    $nama = $_POST["nama"];
    $kelamin = $_POST["jenis_kelamin"];
    $sql = "UPDATE mahasiswa set nim='".$nim."', nama='".$nama."', jenis_kelamin='".$jenis_kelamin."' WHERE id='".$id.'";
    if(mysqli_query($conn, $sql)) {
        echo "Berhasil mengubah data.";
    } else {
        echo "Gagal mengubah data.";
    }
} else {
```

#### D.5. Delete Data

Delete data, atau yang sering disebut dengan DELETE data digunakan untuk menghapus data mahasiswa. Logika berpikir untuk tahap ini:

1. Pada halaman SELECT atau daftar mahasiswa, kolom action, tambahkan sebuah tombol/link yang membawa parameter GET id dari data yang hendak di hapus.
2. Lakukan query untuk DELETE data
3. Berikan notifikasi hapus sudah berhasil diubah

Langkah Kerja:

1. Buka file index.php atau file untuk daftar mahasiswa.
4. Pada bagian perulangan data, kolom terakhir, tambahkan link untuk ke halaman delete-mhs.php. Yang perlu diperhatikan adalah, id data GET yang dikirimkan.

```
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>".$row["nim"]."</td>";
    echo "<td>".$row["nama"]."</td>";
    echo "<td>".$row["jenis_kelamin"]."</td>";
    echo "<td>";
        echo "<a href='form-mhs.php?id=".$row['id']."'>Edit</a>";
        echo "<a href='delete-mhs.php?id=".$row['id']."'>Delete</a>";
    echo "</td>";
    echo "</tr>";
}
```

5. Pada awal halaman, buat koneksi ke database.

```
// open connection
$servername = "127.0.0.1";
$username = "progweb";
$password = "progweb";
$databasename = "php2";
$conn = mysqli_connect($servername, $username, $password, $databasename) or die("koneksi gagal.");
```

6. Lakukan pemeriksaan id pada method GET.
7. Delete mahasiswa dengan id pada parameter GET.

```
if($_GET){
    //DELETE
    $id=$_GET['id'];
    $sql = "DELETE FROM mahasiswa WHERE id='".$id."'";
    if(mysqli_query($conn, $sql)) {
        echo "Berhasil menghapus data. Jumlah data dihapus=".mysqli_affected_rows($conn);
    } else {
        echo "Gagal menghapus data data.";
    }
}
```

8. Lakukan close connection

```
// close connection
mysqli_close($conn);
```

## E. Latihan Terbimbing

Hal mendasar untuk mengelola konten pada sebuah halaman website adalah SELECT, INSERT, UPDATE, dan DELETE. Pada praktikum ini, akan dijelaskan bagaimana mengolah data dari pengguna. Topik yang digunakan untuk penjelasan bagian ini adalah entitas Mahasiswa.

### E.1. Database

Pada pengembangan website menggunakan PHP dan MariaDB/MySQL pada umumnya akan menggunakan xampp bundling. Didalam xampp bundling terdapat phpMyAdmin sebagai website untuk mengelola data pada database. Untuk mengakses phpMyAdmin dapat melalui URL <http://localhost/phpmyadmin>.

Untuk melanjutkan praktikum, buatlah database php2, dengan tabel mahasiswa. Kolom dari tabel mahasiswa dapat dilihat pada gambar dibawah ini.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	nim	varchar(8)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	nama	varchar(50)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	jenis_kelamin	varchar(1)	latin1_swedish_ci		No	None			Change  Drop  More

## E.2. Select Data

Select data, atau yang sering disebut dengan READ data biasanya merupakan tahap paling awal dalam pembuatan sebuah website. Pada bagian ini kita akan mencoba membuat sebuah tabel yang menampilkan data dari entitas mahasiswa. Logika berpikir untuk tahap ini:

6. Buat koneksi ke database
7. Ambil daftar mahasiswa dari database menggunakan query SELECT
8. Periksa apakah ada daftar mahasiswa
9. Apabila ada, buat tabel header
10. Buat perulangan untuk menampilkan setiap baris

Langkah Kerja:

8. Buat sebuah halaman index.php, pada halaman ini, daftar mahasiswa akan ditampilkan pada sebuah tabel.
9. Pada bagian atas halaman, buat koneksi ke database.

```
// open connection
$servername = "127.0.0.1";
$username = "progweb";
$password = "progweb";
$databasename = "php2";
$conn = mysqli_connect($servername, $username, $password, $databasename) or die("Koneksi gagal.");
```

10. Ambil data mahasiswa menggunakan query SELECT.

```
//SELECT
$sql = "SELECT * FROM mahasiswa";
$result = mysqli_query($conn, $sql); //untuk run query
```

11. Periksa apakah ada daftar mahasiswa.

```
if(mysqli_num_rows($result) > 0) {
} else {
    echo "Tabel mahasiswa kosong.";
}
```

12. Buat header untuk tabel daftar mahasiswa. Header yang ditampilkan adalah NIM, nama, jenis kelamin, dan sebuah kolom tambahan untuk edit dan delete.

```

if(mysqli_num_rows($result) > 0) {
    ?>
    <table border=1>
        <thead>
            <th>NIM</th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
            <th>Action</th>
        </thead>
    </table>
    <?php
} else {
    echo "Tabel mahasiswa kosong.";
}
?>

```

13. Buat sebuah perulangan untuk menampilkan setiap baris data menggunakan tag `<tr>` dan `<td>`.

```

<?php
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>".$row["nim"]."</td>";
    echo "<td>".$row["nama"]."</td>";
    echo "<td>".$row["jenis_kelamin"]."</td>";
    echo "<td></td>";
    echo "</tr>";
}
?>

```

14. Hasil yang didapat adalah

NIM	Nama	Jenis Kelamin	Action
71150001	Danny Sebastian	L	
71150002	Maria Nila A	P	

### E.3. Insert Into

Insert into, atau yang sering disebut dengan CREATE data digunakan untuk menambah data mahasiswa. Pada bagian ini kita akan mencoba membuat sebuah form yang digunakan untuk menerima data mahasiswa baru. Logika berpikir untuk tahap ini:

15. Tampilkan form input

16. Pada saat disubmit, Periksa inputan sudah sesuai atau belum.



17. Bila input sudah sesuai, buat koneksi ke database, kemudian isikan data menggunakan query INSERT
18. Berikan notifikasi input sudah berhasil disimpan
19. Bila pada langkah 2 masih ada data yang tidak sesuai, berikan notifikasi kesalahan

Langkah Kerja:

12. Buat halaman form-mhs.php, halaman ini akan menampilkan sebuah form untuk melakukan edit dan insert data mahasiswa.
13. Isikan form dengan input type masing-masing field, berikan parameter action ke file yang akan mengelola input tersebut. Pada kasus ini file yang mengelola input adalah form-mhs.php atau dirinya sendiri.

```
<form action="form-mhs.php" method="POST">
  nim : <input type="number" name="nim" max="8"> <br>
  nama : <input type="text" name="nama" max="50"> <br>
  kelamin : <input type="radio" value="L" name="jenis_kelamin">Laki-laki
            <input type="radio" value="P" name="jenis_kelamin">Perempuan <br>
  <input type="submit">
</form>
```

14. Karena `action="form-mhs.php"` atau dirinya sendiri, maka pada bagian atas, sebelum tag `<html>`, tambahkan open connection.

```
// open connection
$servername = "127.0.0.1";
$username = "progweb";
$password = "progweb";
$databasename = "php2";
$conn = mysqli_connect($servername, $username, $password, $databasename) or die("Koneksi gagal.");
```

15. Selanjutnya, kita melakukan pengecekan, apakah ada data pada variable `$_POST`. Bila ada, maka lakukan proses insert ke database.

```
if($_POST){
  $nim = $_POST["nomer"];
  $nama = $_POST["nama"];
  $kelamin = $_POST["jenis_kelamin"];
  $sql = "INSERT INTO mahasiswa (nim, nama, jenis_kelamin) VALUES ('.$nim.', '$nama.', '$kelamin.')";
  if(mysqli_query($conn, $sql)) {
    echo "Berhasil mengisi data.";
  } else {
    echo "Gagal mengisi data.";
  }
}
```

16. Lakukan close connection

```
// close connection
mysqli_close($conn);
```

#### E.4. Update Data

Update data, atau yang sering disebut dengan UPDATE data digunakan untuk mengubah data mahasiswa. Pada bagian ini kita akan menggunakan form mahasiswa yang sudah dibuat pada tahap sebelumnya. Logika berpikir untuk tahap ini:

17. Pada halaman SELECT atau daftar mahasiswa, kolom action, tambahkan sebuah tombol/link menuju ke halaman form dengan membawa parameter GET id dari data yang hendak di edit.
18. Pada form mahasiswa, apabila ada data GET id, maka tambahkan sebuah `<input type="hidden" value="id">`
19. Pada saat disubmit, Periksa inputan sudah sesuai atau belum.
20. Bila input sudah sesuai, buat koneksi ke database, kemudian isikan data menggunakan query UPDATE dengan WHERE adalah id dari input type hidden
21. Berikan notifikasi input sudah berhasil diubah
22. Bila pada langkah 3 masih ada data yang tidak sesuai, berikan notifikasi kesalahan

Langkah Kerja:

7. Buka file index.php atau file untuk daftar mahasiswa.
8. Pada bagian perulangan data, kolom terakhir, tambahkan link untuk ke halaman form-mhs.php. Yang perlu diperhatikan adalah, id data GET yang dikirimkan.

```
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>".$row["nim"]."</td>";
    echo "<td>".$row["nama"]."</td>";
    echo "<td>".$row["jenis_kelamin"]."</td>";
    echo "<td>";
        echo "<a href='form-mhs.php?id=".$row['id']."'>Edit</a>";
    echo "</td>";
    echo "</tr>";
}
```

9. Buat sebuah percabangan, bila ada data GET, maka proses pengambilan data awal/data yang hendak dirubah. Lakukan pencarian ke database untuk mahasiswa yang memiliki id GET.



```

if($_GET){
    $id = $_GET['id'];
    $sql = "SELECT * FROM mahasiswa WHERE id='".$_GET['id']."'";
    $result = mysqli_query($conn, $sql);
    if(mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_assoc($result);
        $oldNim = $row["nim"];
        $oldNama = $row["nama"];
        $oldKelamin = $row["jenis_kelamin"];
    }
    else {
        echo "Data yang hendak diedit tidak ada.";
    }
}

```

10. Pada bagian `<form>`, tambahkan input hidden untuk id, dan isikan data awal berdasarkan data dari database.

```

<form action="form-mhs.php" method="POST">
    <?php
        if($id != null) {
            ?>
            <input type="hidden" name="id" value="<?php echo $id; ?>">
            <?php
        }
    ?>
    nim : <input type="number" name="nim" max="8" value="<?php echo $oldNim; ?>"> <br>
    nama : <input type="text" name="nama" max="50" value="<?php echo $oldNama; ?>"> <br>
    kelamin : <input type="radio" value="L" name="jenis_kelamin" <?php echo ($oldKelamin=='L')?'checked':''; ?> >Laki-laki
    <input type="radio" value="P" name="jenis_kelamin" <?php echo ($oldKelamin=='P')?'checked':''; ?> >Perempuan <br>
    <input type="submit">
</form>

```

11. Pada bagian INSERT data, buat percabangan untuk UPDATE data.

```

if($_POST){
    if($_POST['id'] != null){
        //update
    } else {
        //insert
        $nim = $_POST["nim"];
        $nama = $_POST["nama"];
        $kelamin = $_POST["jenis_kelamin"];
        $sql = "INSERT INTO mahasiswa (nim, nama, jenis_kelamin) VALUES ('".$_nim."', '".$_nama."', '".$_kelamin."')";
        if(mysqli_query($conn, $sql)) {
            echo "Berhasil mengisi data.";
        } else {
            echo "Gagal mengisi data.";
        }
    }
}

```

12. Tambahkan logika untuk melakukan UPDATE ke database.

```
if($_POST['id'] != null){
    //update
    $nim = $_POST["id"];
    $nim = $_POST["nim"];
    $nama = $_POST["nama"];
    $kelamin = $_POST["jenis_kelamin"];
    $sql = "UPDATE mahasiswa set nim='".$nim."', nama='".$nama."', jenis_kelamin='".$jenis_kelamin."' WHERE id='".$id."'";
    if(mysqli_query($conn, $sql)) {
        echo "Berhasil mengubah data.";
    } else {
        echo "Gagal mengubah data.";
    }
} else {
```

### E.5. Delete Data

Delete data, atau yang sering disebut dengan DELETE data digunakan untuk menghapus data mahasiswa. Logika berpikir untuk tahap ini:

13. Pada halaman SELECT atau daftar mahasiswa, kolom action, tambahkan sebuah tombol/link yang membawa parameter GET id dari data yang hendak di hapus.
14. Lakukan query untuk DELETE data
15. Berikan notifikasi hapus sudah berhasil diubah

Langkah Kerja:

2. Buka file index.php atau file untuk daftar mahasiswa.
16. Pada bagian perulangan data, kolom terakhir, tambahkan link untuk ke halaman delete-mhs.php. Yang perlu diperhatikan adalah, id data GET yang dikirimkan.

```
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>".$row["nim"]."</td>";
    echo "<td>".$row["nama"]."</td>";
    echo "<td>".$row["jenis_kelamin"]."</td>";
    echo "<td>";
        echo "<a href='form-mhs.php?id=".$row['id']."'>Edit</a>";
        echo "<a href='delete-mhs.php?id=".$row['id']."'>Delete</a>";
    echo "</td>";
    echo "</tr>";
}
```

17. Pada awal halaman, buat koneksi ke database.

```
// open connection
$servername = "127.0.0.1";
$username = "progweb";
$password = "progweb";
$databasename = "php2";
$conn = mysqli_connect($servername, $username, $password, $databasename) or die("koneksi gagal.");
```

18. Lakukan pemeriksaan id pada method GET.

19. Delete mahasiswa dengan id pada parameter GET.

```
if($_GET){
    //DELETE
    $id=$_GET['id'];
    $sql = "DELETE FROM mahasiswa WHERE id='".$id."'";
    if(mysqli_query($conn, $sql)) {
        echo "Berhasil menghapus data. Jumlah data dihapus=".mysqli_affected_rows($conn);
    } else {
        echo "Gagal menghapus data data.";
    }
}
```

20. Lakukan close connection

```
// close connection
mysqli_close($conn);
```

## F. Latihan Mandiri

Buatlah 2 buah halaman website untuk mengelola daftar pengguna. Halaman pertama adalah halaman untuk menampilkan daftar pengguna. Halaman kedua adalah form untuk melakukan registrasi pengguna dan update data pengguna. Tambahkan tombol untuk melakukan reset dan submit form. Daftar field dan input type pada kasus ini adalah:

Field	Input Type
Nama Depan	Text
Nama Belakang	Text
Jenis Kelamin	Radio
Tanggal Lahir	Date
Hobi	Checkbox
Email	Email
Alamat	Textarea

Password	Password
Confirmation Password	Password