

# **HANDS-ON LABS 6 PEMROGRAMAN INTERNET + LAB**



**Oleh:**

Martogi Jekson C. Siagian 2105541054

**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2023**

## Hands-On Labs 6

### Tugas

1. Buatlah sebuah aplikasi sederhana berbasis web untuk menangani keperluan pengisian buku tamu (guestbook). Data yang diisikan dalam form adalah sebagai berikut.

Nama

Email

Website

Komentar

Adapun tabel dalam database untuk menyimpan data isian buku tamu tersebut terdiri atas beberapa field sebagai berikut.

`Id        integer        auto increment        primary key`

`Nama    varchar(30)`

`Email   varchar(30)`

`Website varchar(30)`

`Tanggal datetime`

`Komentar text`

Keterangan:

Field "tanggal" digunakan untuk menyimpan informasi tanggal dan waktu kapan komentar tersebut dikirim (tanggal ini secara otomatis membaca tanggal dan waktu ketika komentar buku tamu di-submit). Tambahkan pula pagination untuk data buku tamu yang ditampilkan.

Jawab :

Berikut Beberapa kode yang saya buat untuk program tersebut

## Index.php

```
index.php X delete.php config.php process.php
index.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Buku Tamu</title>
5 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
6 <style>
7     h1,h2 {text-align: center;}
8 </style>
9 </head>
10 <body>
11 <div class="container">
12 <h1 class="mt-4">Buku Tamu</h1>
13
14 <!-- Form untuk menambahkan komentar -->
15 <form action="process.php" method="post">
16 <div class="form-group">
17 <label for="nama">Nama:</label>
18 <input type="text" class="form-control" name="nama" id="nama">
19 </div>
20 <div class="form-group">
21 <label for="email">Email:</label>
22 <input type="text" class="form-control" name="email" id="email">
23 </div>
24 <div class="form-group">
25 <label for="website">Website:</label>
26 <input type="text" class="form-control" name="website" id="website">
27 </div>
28 <div class="form-group">
29 <label for="komentar">Komentar:</label>
30 <textarea class="form-control" name="komentar" id="komentar"></textarea>
31 </div>
32 <button type="submit" class="btn btn-primary">Tambahkan Komentar</button>
33 </form>
34
```

```
index.php X delete.php config.php process.php
index.php
35 <h2 class="mt-4">Data Buku Tamu</h2>
36
37 <?php
38 require('config.php');
39 $items_per_page = 3;
40 $page = isset($_GET['page']) ? $_GET['page'] : 1;
41 $start = ($page - 1) * $items_per_page;
42
43 $query = "SELECT * FROM guestbook ORDER BY Tanggal DESC LIMIT $start, $items_per_page";
44 $result = mysqli_query($conn, $query);
45
46 while ($row = mysqli_fetch_assoc($result)) {
47     echo <div class="card mt-2">;
48     echo <div class="card-body">;
49     echo <h5 class="card-title"> . $row['Nama'] . </h5>;
50     echo <p class="card-text"> . $row['Email'] . </p>;
51     echo <p class="card-text"> . $row['Website'] . </p>;
52     echo <p class="card-text"> . $row['Komentar'] . </p>;
53     echo <p class="card-text"> . Tanggal: . $row['Tanggal'] . </p>;
54     echo <a href="delete.php?id=" . $row['Id'] . " class="btn btn-danger">Hapus</a>;
55     echo </div>;
56     echo </div>;
57 }
58
59 $total_items = mysqli_num_rows(mysqli_query($conn, "SELECT * FROM guestbook"));
60 $total_pages = ceil($total_items / $items_per_page);
61
62 for ($i = 1; $i <= $total_pages; $i++) {
63     echo <a class="btn btn-secondary m-2" href="index.php?page=" . $i . "> . $i . </a>;
64 }
65 ?>
66 </div>
67 </body>
68 </html>
```

Berikut Penjelasan mengenai kode Index.php yang saya buat

1. <!DOCTYPE html>: Ini adalah deklarasi DOCTYPE HTML yang mendefinisikan jenis dokumen HTML yang digunakan oleh halaman web. Dokumen HTML ini mengikuti standar HTML5.

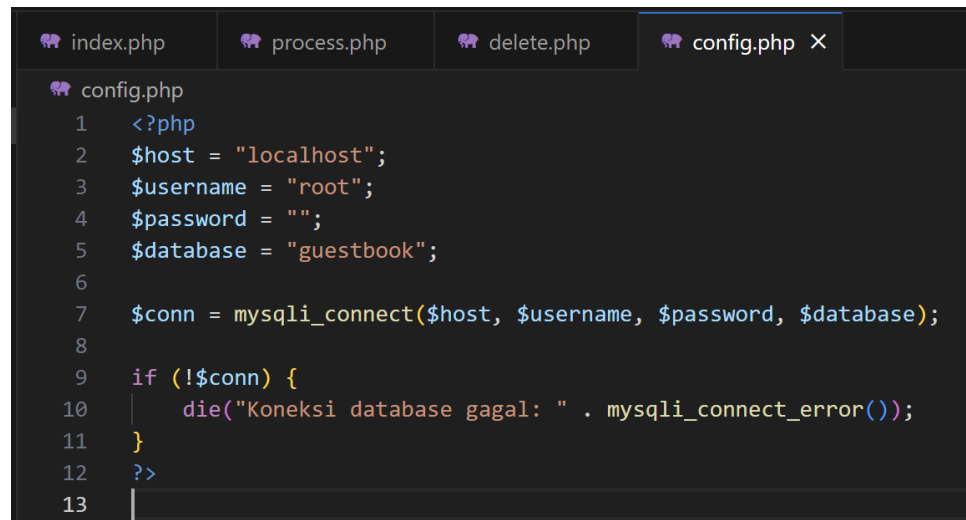
2. `<html>`: Tag HTML yang menandai awal dari dokumen HTML.
3. `<head>`: Bagian ini berisi elemen-elemen yang digunakan untuk mendefinisikan berbagai metadata dokumen seperti judul (title), stylesheet, dan lain-lain.
  - `<title>`: Menentukan judul halaman web yang akan ditampilkan pada tab browser.
  - `<link rel="stylesheet">`: Menghubungkan halaman web dengan stylesheet eksternal dari Bootstrap CSS. Ini digunakan untuk mengatur tampilan dan gaya halaman web dengan bantuan Bootstrap. Saya menggunakan library yang telah disediakan Bootstrap untuk mempermudah mengatur tampilan guestbook yang saya buat
  - `<style>`: Ini adalah blok inline CSS yang menentukan gaya untuk elemen-elemen `<h1>` dan `<h2>` pada halaman.
4. `<body>`: Ini adalah bagian inti dari halaman web yang berisi semua elemen yang akan ditampilkan kepada pengguna.
  - `<div class="container">`: Ini adalah elemen div dengan kelas "container" dari Bootstrap. Itu digunakan untuk mengatur konten agar rapi dan responsif.
  - `<h1>`: Menampilkan judul utama "Buku Tamu" dengan menggunakan kelas "mt-4" yang merupakan kelas margin atas dari Bootstrap.
  - `<form>`: Ini adalah elemen form untuk mengumpulkan data dari pengguna. Atribut action menunjukkan ke file "process.php" yang akan mengolah data saat form disubmit. Metode pengiriman data adalah POST.
    - ✓ `<div class="form-group">`: Ini adalah elemen div dengan kelas "form-group" dari Bootstrap yang digunakan untuk mengelompokkan elemen-elemen formulir bersama-sama.
    - ✓ `<label>`: Label yang menunjukkan deskripsi dari setiap input.

- ✓ `<input>`: Input yang memungkinkan pengguna memasukkan nama, email, website, dan komentar.
- ✓ `<textarea>`: Input teks yang memungkinkan pengguna memasukkan komentar yang lebih panjang.
- ✓ `<button>`: Tombol "Tambahkan Komentar" yang akan mengirimkan data ke "process.php" saat ditekan.
- `<h2>`: Judul kedua yang menampilkan "Data Buku Tamu".
- Kode PHP: Ini adalah blok PHP yang digunakan untuk berinteraksi dengan database. Beberapa poin penting dalam blok PHP ini adalah:
  - ✓ Terhubung ke database dengan menggunakan `require('config.php')`.
  - ✓ Mengambil data dari tabel "guestbook" dalam database.
  - ✓ Menampilkan data dalam kartu (card) dengan menggunakan loop `while`.
  - ✓ Menambahkan tombol "Hapus" untuk setiap entri tamu dengan tautan ke "delete.php".
  - ✓ Menampilkan link halaman pagination sesuai dengan jumlah halaman yang ada.

5. Tag penutup `</html>`: Menandai akhir dari dokumen HTML.

Selain itu, guestbook yang saya buat juga memiliki fitur paginasi untuk mengelola banyaknya komentar yang ditampilkan.

## config.php



```
1 <?php
2 $host = "localhost";
3 $username = "root";
4 $password = "";
5 $database = "guestbook";
6
7 $conn = mysqli_connect($host, $username, $password, $database);
8
9 if (!$conn) {
10     die("Koneksi database gagal: " . mysqli_connect_error());
11 }
12 ?>
13
```

Kode PHP diatas adalah bagian dari sebuah skrip yang digunakan untuk menghubungkan halaman web dengan database MySQL. Berikut penjelasannya

1. \$host, \$username, \$password, dan \$database: Variabel-variabel ini digunakan untuk menyimpan informasi yang diperlukan untuk menghubungkan ke database MySQL.
  - \$host: Menyimpan alamat atau hostname dari server database. Dalam kasus ini, "localhost" digunakan, yang mengindikasikan bahwa database MySQL berjalan di server yang sama dengan aplikasi web.
  - \$username: Menyimpan nama pengguna (username) yang digunakan untuk mengakses database. Dalam kasus ini, "root" digunakan, yang seringkali merupakan nama pengguna dengan akses tertinggi.
  - \$password: Menyimpan kata sandi (password) yang diperlukan untuk otentikasi ke database.
  - \$database: Menyimpan nama database yang akan digunakan dalam skrip ini, yaitu "guestbook."
2. mysqli\_connect(): Ini adalah fungsi yang digunakan untuk membuat koneksi ke database MySQL. Fungsi ini mengambil empat parameter: host, username, password, dan nama database.

- `mysqli_connect($host, $username, $password, $database)` mencoba untuk membuat koneksi ke database MySQL dengan informasi yang telah disimpan di variabel sebelumnya.
3. `if (!$conn)`: Ini adalah blok kondisional yang digunakan untuk memeriksa apakah koneksi ke database berhasil atau tidak.
- jika koneksi gagal (nilai `$conn` adalah false), maka blok die akan dijalankan.
  - `die("Koneksi database gagal: " . mysqli_connect_error())` digunakan untuk mengakhiri eksekusi skrip dan menampilkan pesan kesalahan berupa "Koneksi database gagal" diikuti oleh pesan kesalahan yang diberikan oleh `mysqli_connect_error()`. Pesan kesalahan ini akan berguna untuk debug jika ada masalah dalam koneksi database.

Dengan demikian, kode ini bertujuan untuk menghubungkan halaman web ke database MySQL yang memiliki nama "guestbook" di server localhost. Jika koneksi gagal, pesan kesalahan akan ditampilkan untuk memberi tahu bahwa koneksi database tidak berhasil.

## Process.php

```
process.php X
process.php
1  <?php
2  require('config.php');
3
4  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5      $nama = $_POST['nama'];
6      $email = $_POST['email'];
7      $website = $_POST['website'];
8      $komentar = $_POST['komentar'];
9
10     $query = "INSERT INTO guestbook (Nama, Email, Website, Tanggal, Komentar)
11             VALUES ('$nama', '$email', '$website', NOW(), '$komentar')";
12     $result = mysqli_query($conn, $query);
13
14     if ($result) {
15         header('Location: index.php');
16     } else {
17         echo "Error: " . mysqli_error($conn);
18     }
19 }
20 ?>
```

Berikut penjelasan untuk kode ini

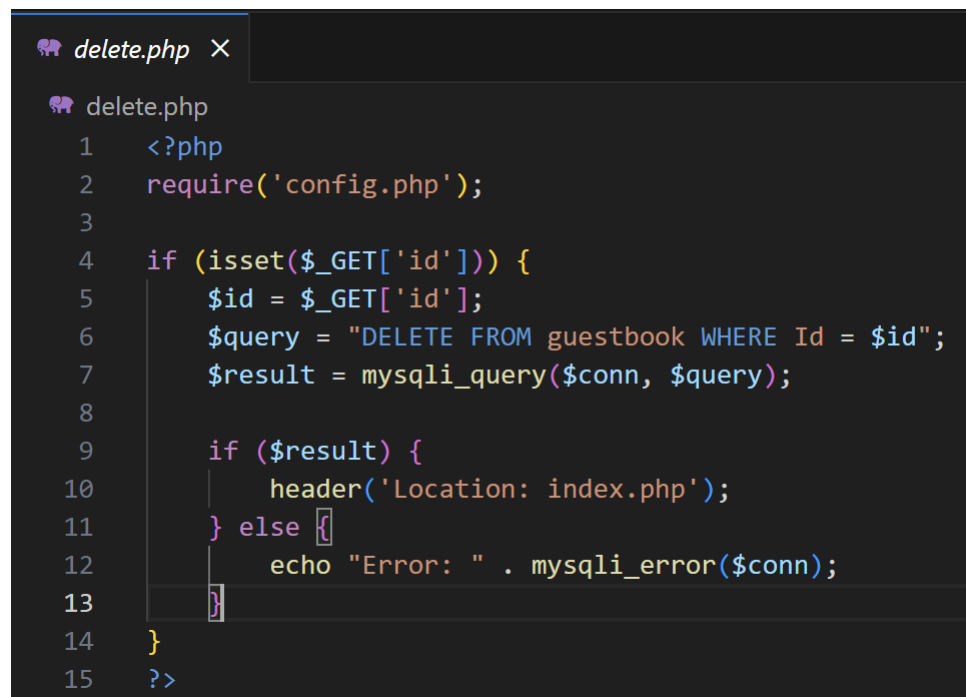
1. `require('config.php')`: Ini adalah pernyataan `require` yang digunakan untuk memasukkan file `config.php`. File ini kemungkinan berisi informasi koneksi ke database, seperti host, nama pengguna, kata sandi, dan nama database. Ini memastikan bahwa Anda memiliki koneksi database yang sudah diatur sebelum melanjutkan.
2. `if ($_SERVER['REQUEST_METHOD'] === 'POST')`: Ini adalah blok kondisional yang memeriksa apakah permintaan (request) yang diterima adalah dengan metode HTTP POST. Kode di dalam blok ini hanya akan dieksekusi jika permintaan itu berasal dari formulir yang dikirim menggunakan POST.
3. Pengambilan Data Form:
  - `$nama`, `$email`, `$website`, dan `$komentar`: Variabel ini digunakan untuk menyimpan data yang diambil dari formulir HTML dengan menggunakan metode POST. Ini adalah nilai-nilai yang diisi oleh pengguna melalui formulir.
4. SQL Query:
  - `$query`: Ini adalah variabel yang berisi pernyataan SQL INSERT yang akan memasukkan data ke dalam tabel "guestbook" di database. Nilai-nilai yang diambil dari formulir digunakan dalam pernyataan SQL ini.
  - Pada `$query`, terdapat penggunaan `NOW()` untuk mengisi kolom "Tanggal" dengan nilai tanggal saat ini, yang akan mencatat tanggal dan waktu saat komentar ditambahkan ke buku tamu.
5. Eksekusi Query:
  - `$result`: Ini adalah variabel yang digunakan untuk mengeksekusi pernyataan SQL yang telah dibuat. Hasil eksekusi disimpan dalam variabel ini.
6. Pengolahan Hasil:
  - Blok `if ($result)` akan dijalankan setelah eksekusi query. Jika query berhasil dieksekusi (misalnya, data berhasil disimpan), maka pengguna akan diarahkan kembali ke halaman "index.php" dengan menggunakan `header('Location: index.php')`.



- Jika query gagal (misalnya, ada kesalahan dalam query SQL), pesan kesalahan akan ditampilkan dengan `echo "Error: " . mysqli_error($conn)`. Ini akan membantu Anda dalam debug jika ada masalah dengan query SQL.

Jadi, kode ini bertugas untuk mengambil data dari formulir POST, menyimpannya ke dalam tabel database "guestbook," dan mengarahkan kembali pengguna ke halaman "index.php" setelah data berhasil disimpan. Jika ada masalah dalam proses penyimpanan data, pesan kesalahan akan ditampilkan.

### delete.php



```

1  <?php
2  require('config.php');
3
4  if (isset($_GET['id'])) {
5      $id = $_GET['id'];
6      $query = "DELETE FROM guestbook WHERE Id = $id";
7      $result = mysqli_query($conn, $query);
8
9      if ($result) {
10         header('Location: index.php');
11     } else {
12         echo "Error: " . mysqli_error($conn);
13     }
14 }
15 ?>

```

Kode PHP diatas adalah bagian dari skrip yang bertujuan untuk menghapus entri dari tabel database "guestbook" berdasarkan ID yang diberikan melalui parameter GET (query string). Di sini, saya akan menjelaskan setiap bagian dari kode ini:

1. `require('config.php')`: Ini adalah pernyataan `require` yang digunakan untuk memasukkan file `config.php`. File ini kemungkinan berisi informasi koneksi ke database, seperti host, nama pengguna, kata sandi,

dan nama database. Ini memastikan bahwa Anda memiliki koneksi database yang sudah diatur sebelum melanjutkan.

2. `if (isset($_GET['id']))`: Ini adalah blok kondisional yang memeriksa apakah parameter GET dengan nama "id" telah diset. Parameter ini seharusnya berisi ID yang akan digunakan untuk mengidentifikasi entri yang akan dihapus.
3. Pengambilan ID:
  - `$id`: Variabel ini digunakan untuk menyimpan nilai ID yang diterima dari parameter GET "id".
4. SQL Query:
  - `$query`: Ini adalah variabel yang berisi pernyataan SQL DELETE yang akan menghapus entri dari tabel "guestbook" berdasarkan ID yang sesuai. Nilai ID yang diambil dari parameter GET digunakan dalam pernyataan SQL ini.
5. Eksekusi Query:
  - `$result`: Variabel ini digunakan untuk mengeksekusi pernyataan SQL yang telah dibuat. Hasil eksekusi disimpan dalam variabel ini.
6. Pengolahan Hasil:
  - Blok `if ($result)` akan dijalankan setelah eksekusi query. Jika query berhasil dieksekusi (misalnya, entri berhasil dihapus), pengguna akan diarahkan kembali ke halaman "index.php" dengan menggunakan `header('Location: index.php')`.
  - Jika query gagal (misalnya, ada kesalahan dalam query SQL), pesan kesalahan akan ditampilkan dengan `echo "Error: " . mysqli_error($conn)`. Ini akan membantu Anda dalam debug jika ada masalah dengan query SQL.

Jadi, kode ini bertugas untuk menghapus entri dari tabel "guestbook" berdasarkan ID yang diberikan melalui parameter GET, dan kemudian mengarahkan kembali pengguna ke halaman "index.php" setelah entri berhasil dihapus.

## Query Sql

```
CREATE TABLE guestbook (  
    Id INT AUTO_INCREMENT PRIMARY  
KEY,  
    Nama VARCHAR(30) ,  
    Email VARCHAR(30) ,  
    Website VARCHAR(30) ,  
    Tanggal DATETIME,  
    Komentar TEXT  
);
```

Kode ini adalah pernyataan SQL yang digunakan untuk membuat sebuah tabel bernama "guestbook" dalam sebuah database. Tabel ini memiliki beberapa kolom dengan tipe data yang berbeda:

1. Id adalah kolom dengan tipe data INT (bilangan bulat) yang memiliki sifat AUTO\_INCREMENT, yang berarti nilainya akan otomatis bertambah satu setiap kali sebuah entri baru ditambahkan. Kolom ini juga ditetapkan sebagai PRIMARY KEY, yang digunakan untuk mengidentifikasi setiap entri secara unik.
2. Nama adalah kolom dengan tipe data VARCHAR(30), yang digunakan untuk menyimpan nama tamu dengan panjang maksimum 30 karakter.
3. Email adalah kolom dengan tipe data VARCHAR(30), yang digunakan untuk menyimpan alamat email tamu dengan panjang maksimum 30 karakter.
4. Website adalah kolom dengan tipe data VARCHAR(30), yang digunakan untuk menyimpan alamat situs web tamu dengan panjang maksimum 30 karakter.
5. Tanggal adalah kolom dengan tipe data DATETIME, yang digunakan untuk menyimpan tanggal dan waktu ketika komentar ditambahkan ke buku tamu.
6. Komentar adalah kolom dengan tipe data TEXT, yang digunakan untuk menyimpan komentar tamu. Kolom ini bisa menampung teks yang lebih panjang dibandingkan dengan VARCHAR.

Berikut Hasil Tampilan dari guestbook yang saya buat

Buku Tamu

Name:

Email:

Website:

Komentar:

[Tambahkan Komentar](#)

Data Buku Tamu

Martogi Jekson C. Siagian  
martogisiagian17@gmail.com  
togi.com  
buku nya sangat bagus  
Tanggal: 2023-10-25 23:19:30  
[Hapus](#)

1 2