

PERTEMUAN XI

PEMROGRAMAN WEB

PHP LANJUTAN

Izmy Alwiah Musdar
izmyalwiah@gmail.com

CONTENT

01

SESSION

02

COOKIE

03

PENANGANAN FORM

04

PENANGANAN DATABASE

TUJUAN PEMBELAJARAN

- Mahasiswa mampu menggunakan session dan cookie untuk penyimpanan data.
- Mahasiswa mampu menggunakan variable superglobal `$_GET`, `$_POST`, dan `$_FILES` untuk pemrosesan data menggunakan form.
- Mahasiswa mampu membuat skrip PHP yang memungkinkan halaman web berinteraksi dengan database

SESSION

- Digunakan untuk menyimpan nilai.
- Sebuah session dapat diakses oleh file lain selama session belum dihapus atau browser belum ditutup.
- Biasanya digunakan pada pembuatan login pengguna, untuk mengecek apakah pengguna sudah login atau belum.
- Pembuatan session menggunakan variabel superglobal `$_SESSION[]` dengan sebelumnya menuliskan `session_start()`.
- Menghapus session menggunakan fungsi `session_destroy()`.

BUAT SESSION

```
<?php
    session_start();
    $_SESSION['username'] = "Admin";
    $_SESSION['password'] = "Rahasia";

    echo "Session telah di buat,
    <a href='6_2_session_cek.php'> Cek Session </a>";
?>
```

6_1_session.php

CEK SESSION

```
<?php
    session_start();

    if(!empty($_SESSION['username']) and !empty($_SESSION['password']))
    {
        echo "Username: ".$_SESSION['username'];
        echo "<br> Password: ".$_SESSION['password'];
        echo "<br> <a href='6_3_session_hapus.php'> Hapus Session </a>";
    } else{
        echo "Session telah dihapus";
        echo "<br> <a href='6_1_session.php'>Buat Session</a>";
    }

?>
```

HAPUS SESSION

```
<?php
    session_start();

    session_destroy();

    echo "Session telah dihapus,
    <a href='6_2_session_cek.php'> Cek Session </a>";
?>
```


COOKIE

- Serupa dengan session, namun data cookie tidak terhapus ketika browser ditutup.
- Cookie terhapus ketika batas waktu berakhir. Batas waktu cookie dinyatakan dalam detik.
- Membuat cookie dengan menggunakan fungsi `setcookie()`.
- Cookie ditampilkan dengan menggunakan variabel superglobal `$_COOKIE[nama_cookie]`.

```
setcookie(nama cookie, nilai, batas akhir).
```


MEMBUAT COOKIE

```
<?php
    $batas = time() + 30;

    setcookie("user", "admin", $batas);

    echo "COOKIE telah dibuat, buka file 6_5_cookie_cek.php
    sebelum dan setelah 30 detik";
    echo "<br><a href='6_5_cookie_cek.php'>cek cookie</a>"
?>
```

6_4_cookie.php

MENGECEK COOKIE

```
<?php
    if (isset($_COOKIE["user"])) {
        echo "Isi Cookie : ".$_COOKIE["user"];
    } else {
        echo "Cookie berakhir";
    }
?>
```

PENANGANAN FORM

- PHP memungkinkan interaksi antara aplikasi web dan pengguna.
- PHP menggunakan variable superglobal untuk pengiriman data dari form, yaitu :
 - ✓ `$_GET`
 - ✓ `$_POST`
 - ✓ `$_FILES`

PENANGANAN FORM

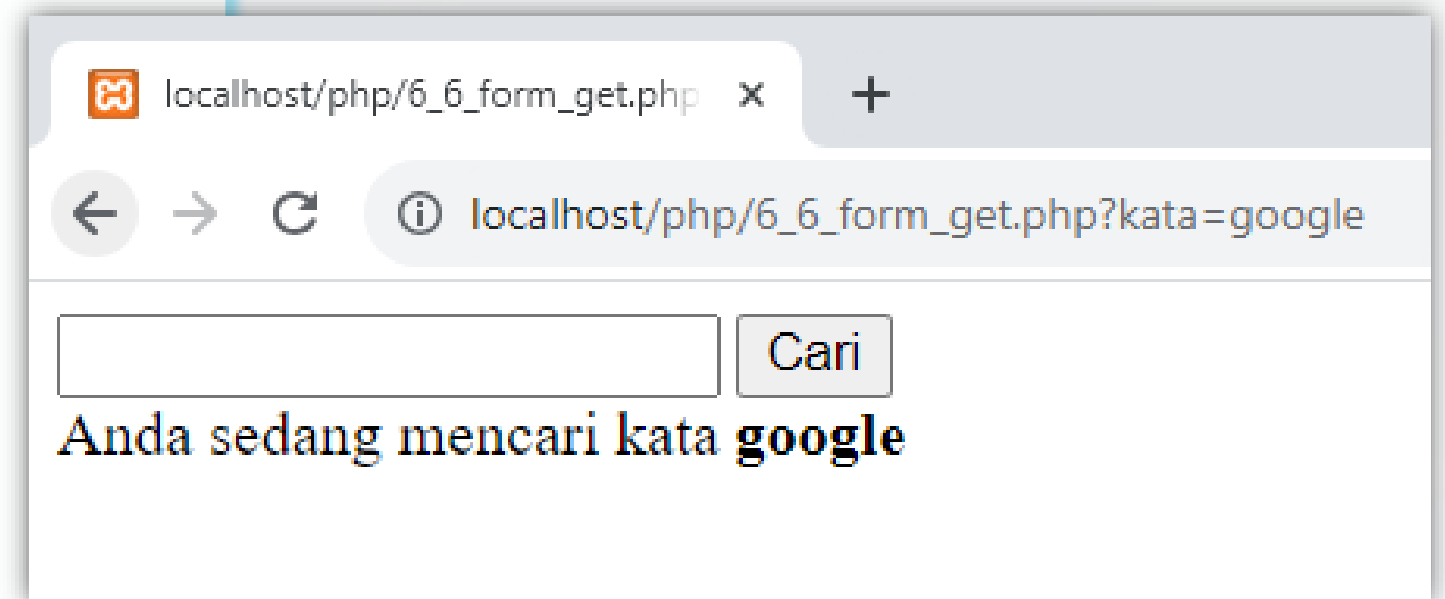
\$_GET

- Menyimpan data array yang dikirim oleh form dengan method GET melalui URL.
- Data tampil pada URL sehingga tidak disarankan untuk mengirim data yang sensitif seperti password.

PENANGANAN FORM : GET

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <body>
    <form method="get" action="">
      <input type="text" name="kata">
      <input type="submit" value="Cari">
    </form>

    <?php
      if(isset($_GET['kata'])){
        echo "Anda sedang mencari kata <b>$_GET[kata]</b>";
      }
    ?>
  </body>
</html>
```



PENANGANAN FORM

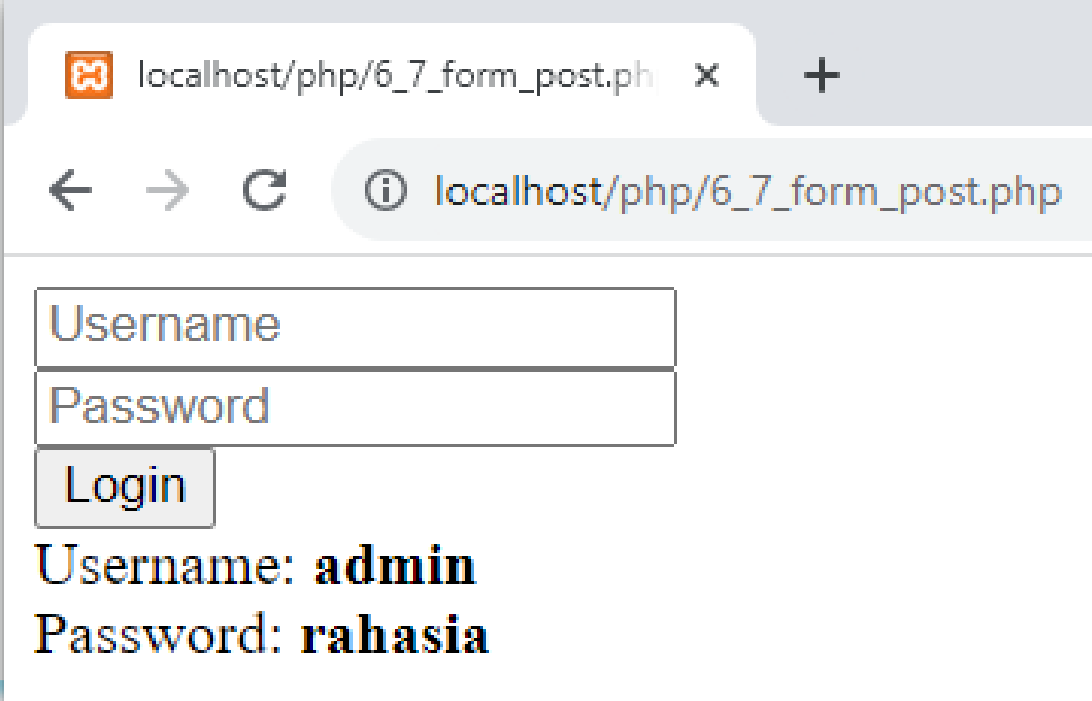
`$_POST`

- Menyimpan data array yang dikirim oleh form dengan method POST melalui melalui HTTP POST.
- Data yang dikirim tidak tampak pada URL.

PENANGANAN FORM : POST

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <body>
    <form method="post" action="">
      <input type="text" placeholder="Username" name="username"><br>
      <input type="password" placeholder="Password" name="password"><br>
      <input type="submit" value="Login">
    </form>

    <?php
      if($_SERVER["REQUEST_METHOD"] == "POST"){
        echo "Username: <b>$_POST[username]</b><br>";
        echo "Password: <b>$_POST[password]</b><br>";
      }
    ?>
  </body>
</html>
```



A screenshot of a web browser window. The address bar shows 'localhost/php/6_7_form_post.php'. The page displays the output of the PHP script: 'Username: admin' and 'Password: rahasia'. The form fields 'Username' and 'Password' are visible above the output, and a 'Login' button is also present.

PENANGANAN FORM

\$_FILES

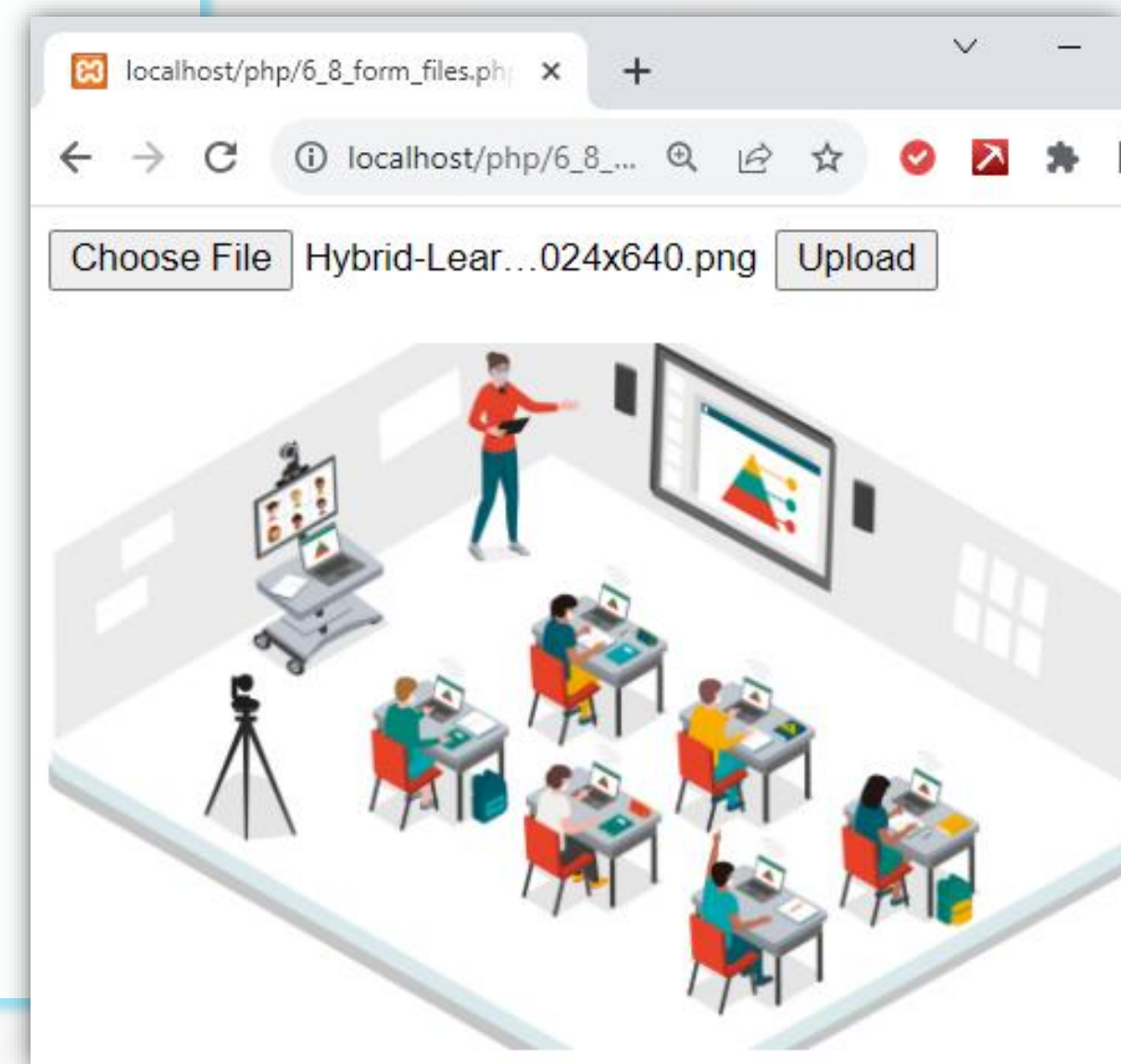
- Menyimpan data yang dikirim dari input berbentuk tipe file.
- Digunakan untuk proses upload File.
- Agar file dapat terupload, pada tag form tambahkan atribut `enctype="multipart/form-data"`.

PENANGANAN FORM : FILES

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <input type="file" name="foto">
      <input type="submit" value="Upload">
    </form>

    <?php
      if($_SERVER["REQUEST_METHOD"] == "POST"){
        $namafile = $_FILES['foto']['name'];
        $lokasifile = $_FILES['foto']['tmp_name'];

        if($namafile != ""){
          move_uploaded_file($lokasifile, "file/".$namafile);
          echo "<br><img src='file/$namafile' width=' 400'>";
        }
      }
    ?>
  </body>
</html>
```



PENANGANAN DATABASE

- Tujuan utama penggunaan PHP adalah untuk penanganan database.
- Data yang diproses oleh PHP dapat disimpan dalam database dan digunakan pada halaman web.
- PHP menggunakan beberapa fungsi untuk penanganan database.
- Penangan DB dengan PHP dapat menggunakan MySQLi Extension atau PHP Data Objects.

PENANGANAN DATABASE

```
$conn = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);  
if(mysqli_connect_erro()) {  
    echo "Koneksi gagal: ".mysqli_connect_error();  
}  
mysqli_close($conn);
```

MySQLi
Procedural

```
$conn = new mysqli($dbhost,$dbuser,$dbpass,$dbname);  
if($conn->connect_error) {  
    echo "Koneksi gagal: ".$conn->connect_error;  
}  
  
$conn->close();
```

MySQLi OO

```
$conn = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser, $dbpass);  
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
$conn = null;
```

PDO

Fungsi	Keterangan
<code>mysqli_affected_row(koneksi)</code>	Menghasilkan jumlah baris yang terpengaruh oleh operasi MySQL sebelumnya.
<code>mysqli_autocommit(koneksi, mode)</code>	Mengeset apakah modifikasi database langsung diproses atau tidak.
<code>mysqli_change_user(koneksi, username, password, dbname)</code>	Mengubah user pada koneksi database.
<code>mysqli_character_set_name(koneksi)</code>	Menghasilkan karakter default pada koneksi database.
<code>mysqli_close(koneksi)</code>	Menutup koneksi database yang sedang terbuka.
<code>mysqli_commit(koneksi)</code>	Memproses transaksi database saat ini.
<code>mysqli_connect_errno()</code>	Menghasilkan kode error dari koneksi terakhir.
<code>mysqli_connect_error()</code>	Menghasilkan deskripsi error dari koneksi terakhir.
<code>mysqli_connect(host, username, password, dbname, port, socket)</code>	Membuka koneksi baru ke MySQL server.
<code>mysqli_data_seek(result, offset)</code>	Menuju baris tertentu dari data yang dihasilkan.
<code>mysqli_debug(message)</code>	Debug operasi database
<code>mysqli_dump_debug_info(koneksi)</code>	Menghasilkan info debug ke log.

Fungsi	Keterangan
mysqli_errno(koneksi)	Menghasilkan kode error terakhir dari beberapa operasi database.
mysqli_error_list(koneksi)	Menghasilkan daftar error dari beberapa operasi database.
mysqli_error(koneksi)	Menghasilkan deskripsi error terakhir dari beberapa operasi database.
mysqli_fetch_all(result, result_type)	Menghasilkan array asosiatif, array numerik atau keduanya dari semua hasil operasi database. Result_type dapat diisi MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH.
mysqli_fetch_array(result, result_type)	Menghasilkan array asosiatif, array numerik atau keduanya dari hasil operasi database.
mysqli_fetch_assoc(result)	Menghasilkan array asosiatif dari hasil operasi database.
mysqli_fetch_field_direct(result, field_number)	Menghasilkan meta-data sebuah field dari hasil operasi database, sebagai sebuah objek.
mysqli_fetch_field(result)	Menghasilkan field berikutnya dari hasil operasi database, sebagai sebuah objek.
mysqli_fetch_fields(result)	Menghasilkan array objek yang mempresentasikan field dari hasil operasi database.
mysqli_fetch_lengths(result)	Menghasilkan jumlah kolom dari baris saat ini.
mysqli_fetch_object(result, classname, parameter)	Menghasilkan baris dari hasil operasi database sebagai sebuah objek.
mysqli_fetch_row(result)	Menghasilkan array numerik dari satu baris hasil operasi database.

Fungsi	Keterangan
<code>mysqli_field_count(koneksi)</code>	Menghasilkan jumlah kolom dari query sebelumnya.
<code>mysqli_field_seek(result, field_number)</code>	Menuju field tertentu.
<code>mysqli_field_tell(result)</code>	Menghasilkan posisi field saat ini.
<code>mysqli_get_client_info(koneksi)</code>	Menghasilkan versi client library MySQL.
<code>mysqli_get_connection_stats(koneksi)</code>	Menghasilkan statistik mengenai koneksi client.
<code>mysqli_get_host_info(koneksi)</code>	Menghasilkan nama host server MySQL dan tipe koneksi.
<code>mysqli_get_proto_info(koneksi)</code>	Menghasilkan versi protokol MySQL.
<code>mysqli_get_server_info(koneksi)</code>	Menghasilkan versi server MySQL.
<code>mysqli_get_server_version(koneksi)</code>	Menghasilkan versi server MySQL sebagai integer.
<code>mysqli_info(koneksi)</code>	Menghasilkan informasi tentang query terakhir yang dieksekusi.
<code>mysqli_init()</code>	Menginisialisasi MySQLi dan menghasilkan resource untuk digunakan dengan <code>mysqli_real_connection()</code> .
<code>mysqli_insert_id(koneksi)</code>	Menghasilkan id yang dihasilkan otomatis dari query terakhir.
<code>mysqli_more_results(koneksi)</code>	Mengecek apakah ada hasil dari beberapa query.
<code>mysqli_multi_query(koneksi, query)</code>	Memproses beberapa query.

Fungsi	Keterangan
mysqli_next_result(koneksi)	Mempersiapkan hasil berikutnya dari mysqli_multi_query().
mysqli_num_fields(result)	Menghasilkan jumlah field dari hasil operasi database.
mysqli_num_rows(result)	Menghasilkan jumlah baris dari hasil operasi database.
mysqli_options(koneksi, option, value)	Menambahkan pengaturan pada koneksi database.
mysqli_ping(koneksi)	Mengetes koneksi server atau mencoba koneksi ulang jika koneksi bermasalah.
mysqli_query(koneksi, query, resultmode)	Memproses query ke database. Resultmode dapat diisi MYSQLI_USE_RESULT, MYSQLI_STORE_RESULT.
mysqli_real_connect(koneksi, host, username, password, dbname, port, socket, flag)	Membuka koneksi baru ke MySQL server.
mysqli_real_escape_string(koneksi, escapestring)	Meloloskan karakter khusus pada string untuk digunakan pada perintah SQL.
mysqli_select_db(koneksi, dbname)	Mengubah database yang terkoneksi.

MEMBUAT DATABASE

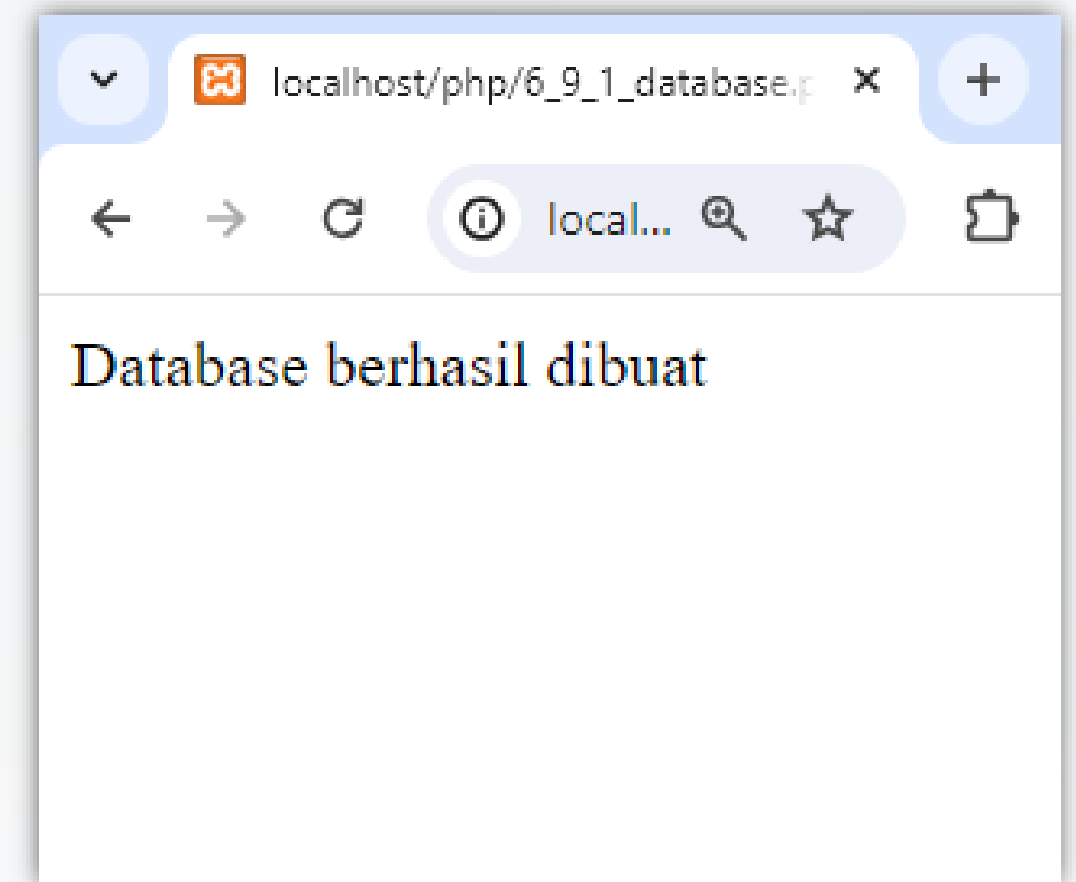
```
<?php

$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$dbname = "db_seminar";

//Membuat koneksi
$conn = mysqli_connect($dbhost,$dbuser,$dbpass);

if(mysqli_connect_errno()){
    echo "Koneksi gagal: ".mysqli_connect_error();
}
//query membuat database
$sql = "CREATE DATABASE $dbname";
if (mysqli_query($conn, $sql)) {
    echo "Database berhasil dibuat";
} else {
    echo "Gagal membuat database: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```



MEMBUAT TABEL

```
<?php

$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$dbname = "db_seminar";

//Membuat koneksi
$conn = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);

if(mysqli_connect_errno()){
    echo "Koneksi gagal: ".mysqli_connect_error();
}
//query membuat tabel
$sql = "CREATE TABLE Peserta (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(30) NOT NULL,
    noHP VARCHAR(13) NOT NULL,
    email VARCHAR(50)
)";

if (mysqli_query($conn, $sql)) {
    echo "Table Peserta berhasil dibuat";
} else {
    echo "Gagal membuat tabel: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

```
<?php
```

```
$dbhost = "localhost";  
$dbuser = "root";  
$dbpass = "";  
$dbname = "db_seminar";
```

```
//Membuat koneksi
```

```
$conn = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);
```

```
if(mysqli_connect_errno()){
```

```
    echo "Koneksi gagal: ".mysqli_connect_error();
```

```
}
```

```
//query menambah data
```

```
$sql = "INSERT INTO Peserta (nama, noHP, email)
```

```
VALUES ('Sakinah', '081234567891', 'sakinah@contoh.com');"
```

```
if (mysqli_query($conn, $sql)) {
```

```
    echo "Data berhasil ditambahkan";
```

```
} else {
```

```
    echo "Gagal menambah data: " . $sql . "<br>" . mysqli_error($conn);
```

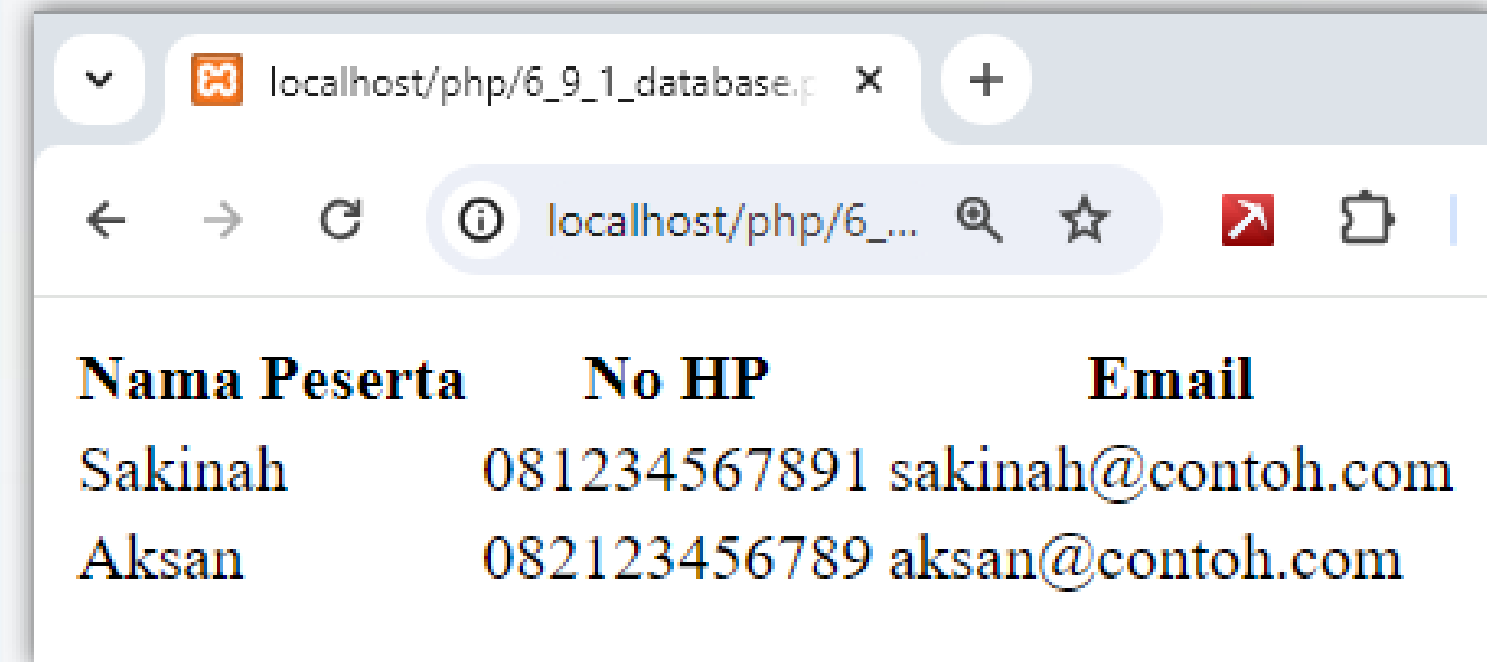
```
}
```

```
mysqli_close($conn);
```

```
?>
```

```
<?php
...
if(mysqli_connect_errno()){
    echo "Koneksi gagal: ".mysqli_connect_error();
}
//query mengambil data dari tabel
$sql = "SELECT * FROM Peserta";
$result = mysqli_query($conn, $sql);

//menampilkan data
echo "<table>";
echo "<tr>";
echo "<th>Nama Peserta</th>";
echo "<th>No HP</th>";
echo "<th>Email</th>";
echo "</tr>";
while($row = mysqli_fetch_array($result)){
    echo "<tr>";
    echo "<td>$row[nama]</td>";
    echo "<td>$row[noHP]</td>";
    echo "<td>$row[email]</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($conn);
?>
```



The screenshot shows a web browser window with a single tab titled 'localhost/php/6_9_1_database.p'. The address bar shows 'localhost/php/6_...'. The page content displays a table with three columns: 'Nama Peserta', 'No HP', and 'Email'. The table contains two rows of data: Sakinah with phone number 081234567891 and email sakinah@contoh.com, and Aksan with phone number 082123456789 and email aksan@contoh.com.

Nama Peserta	No HP	Email
Sakinah	081234567891	sakinah@contoh.com
Aksan	082123456789	aksan@contoh.com

**TERIMA
KASIH**

