

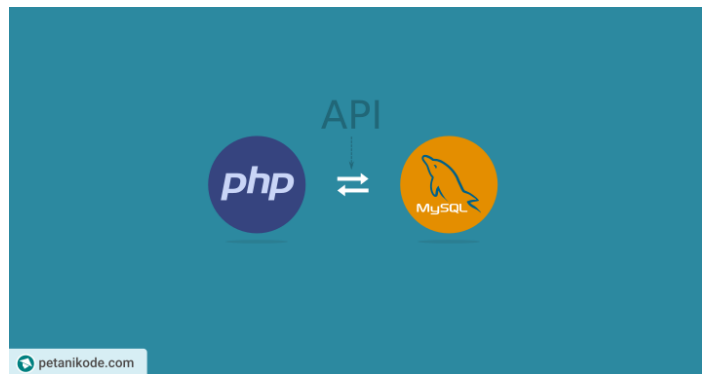
# Modul Praktikum

## Aplikasi CRUD dengan PHP dan MySQL

### MK Basis Data dan Sistem Basis Data 2024

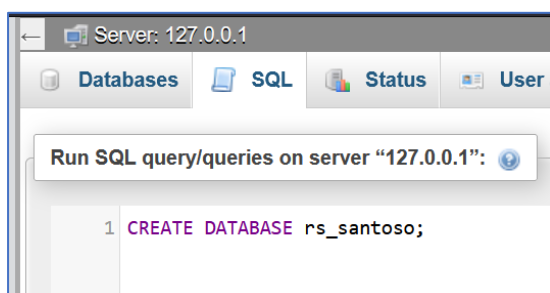
Tutorial Oleh: PetaniKode | Disusun Kembali Oleh: Liptia Venica

Pada kesempatan ini, kita akan belajar menggunakan PHP dan MySQL untuk membuat aplikasi berbasis web sederhana. MySQL berperan sebagai tempat menyimpan data (basis data) sedangkan PHP adalah bahasa pemrograman yang digunakan untuk membuat aplikasi yang bertugas mengatur proses pengelolaan data tersebut. PHP dan MySQL dapat saling berhubungan melalui perantara API (*Application Programming Interface*). Pada praktikum ini, kita akan menggunakan API **mysqli**.

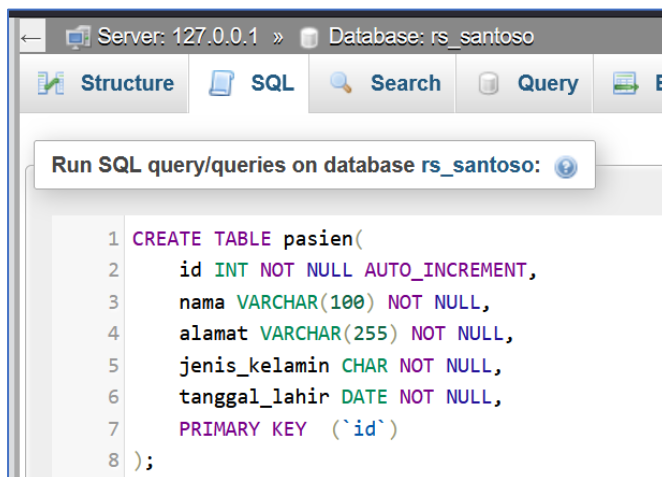


## Menyiapkan Database

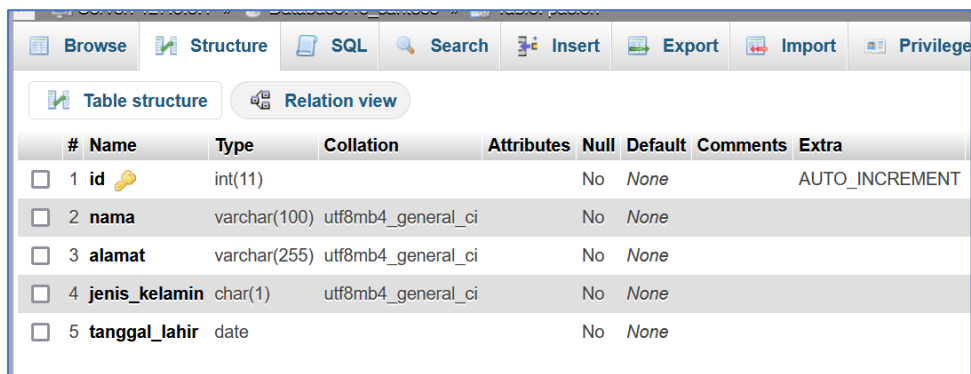
Pertama, mari kita membuat database bernama `rs_santoso`. Pada tahap ini, database tidak dirancang terlebih dahulu karena kita hanya akan praktik dengan 1 tabel saja. Nantinya ketika akan membuat database, langkah awal yang harus dilakukan adalah merancang databasenya terlebih dahulu melalui proses ERD, dikonversi ke skema relasional, dan terakhir normalisasi.



Setelah itu, buatlah tabel di dalam database tersebut dengan nama `pasien`.



```
1 CREATE TABLE pasien(  
2     id INT NOT NULL AUTO_INCREMENT,  
3     nama VARCHAR(100) NOT NULL,  
4     alamat VARCHAR(255) NOT NULL,  
5     jenis_kelamin CHAR NOT NULL,  
6     tanggal_lahir DATE NOT NULL,  
7     PRIMARY KEY (`id`)  
8 );
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 nama	varchar(100)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	3 alamat	varchar(255)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	4 jenis_kelamin	char(1)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	5 tanggal_lahir	date			No	None		

Kemudian kita isi dengan sebuah data seperti berikut.



```
1 INSERT INTO pasien (nama, alamat, jenis_kelamin, tanggal_lahir)  
2 VALUES ('River Jusuf', 'Kebayoran Lama Baru, RT 1 RW 20, Jakarta Tenggara', 'L', '1994-12-31');  
3
```

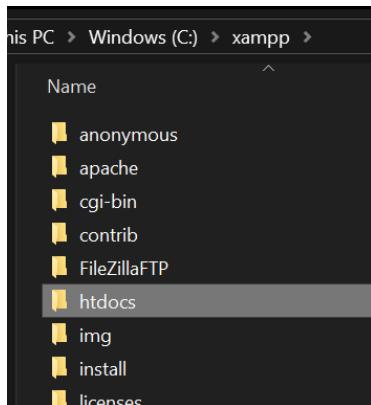
	id	nama	alamat	jenis_kelamin	tanggal_lahir
<input type="checkbox"/>	1	River Jusuf	Kebayoran Lama Baru, RT 1 RW 20, Jakarta Tenggara	L	1994-12-31

Database sudah siap dan kita siap untuk masuk ke tahap pembuatan program.

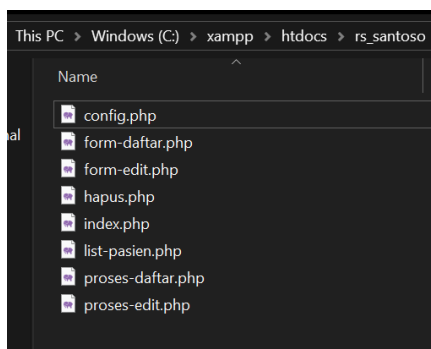
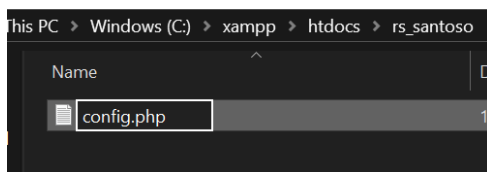
## Menyiapkan Direktori Proyek untuk Aplikasi

Kita akan menggunakan direktori **htdocs** yang ada di XAMPP untuk menyimpan direktori proyek aplikasi web kita. Folder **htdocs** adalah default root directory atau home directory pada XAMPP. Seluruh data website php atau html (kecuali database) harus

ditempatkan di dalam folder ini, agar website pada server yang berjalan dapat diakses. Lokasi folder ini biasanya ada di **Windows C > xampp > htdocs**.



Di dalam folder htdocs, buatlah sebuah folder bernama **rs\_santoso** kemudian isi folder tersebut dengan 8 (delapan) file PHP kosong seperti berikut (file PHP dapat Anda buat manual atau melalui *script* yang dijalankan di command prompt/CMD).



Keterangan:

- **config.php** — untuk menyimpan konfigurasi database;
- **index.php** — halaman utama;
- **list-pasien.php** — halaman untuk menampilkan data pasien;
- **form-daftar.php** — halaman formulir pendaftaran;
- **proses-daftar.php** — skrip yang memproses pendaftaran;
- **form-edit.php** — form untuk edit data pasien;
- **proses-edit.php** — skrip untuk memproses edit/update;
- **hapus.php** — skrip untuk menghapus data di tabel;

## HTML, CSS, dan PHP

Sebelum lanjut ke tahap berikutnya, pastikan Anda mempelajari terlebih dahulu dasar HTML dan juga CSS. Berikut referensi yang dapat Anda gunakan.

<https://www.petanikode.com/tutorial/css/>

<https://www.petanikode.com/tutorial/html/>

<https://www.codepolitan.com/course/intro/basic-html-dan-css/>

<https://www.w3schools.com/html/>

<https://www.w3schools.com/css/>

HTML dan CSS ini nantinya akan berkolaborasi dengan PHP untuk membangun sebuah aplikasi web. HTML, CSS, dan PHP bekerja sama untuk membangun website, namun mereka memiliki peran yang berbeda.

**HTML (HyperText Markup Language):** Merupakan fondasi atau struktur dasar website. HTML menggunakan tag untuk mendefinisikan konten halaman web, seperti heading, paragraf, gambar, dan link. Bayangkan HTML seperti kerangka website yang menentukan letak konten.

**CSS (Cascading Style Sheets):** Berfokus pada presentasi atau tampilan website. CSS mengatur gaya visual elemen HTML, seperti font, warna, background, dan layout. Dengan CSS, Anda dapat membuat website terlihat menarik dan terstruktur. Pikirkan CSS sebagai cat dan perabot yang membuat kerangka HTML menjadi website yang indah.

**PHP (Hypertext Preprocessor):** Adalah bahasa pemrograman server-side. Tidak seperti HTML dan CSS yang diinterpretasikan oleh browser, PHP dijalankan di server sebelum dikirim ke browser. PHP membuat website menjadi dinamis. Anda dapat menggunakan PHP untuk:

- Menampilkan konten yang berbeda berdasarkan pengguna atau situasi.
- Mengolah data dari form.
- Berinteraksi dengan database untuk mengambil dan menyimpan informasi.

## Membuat Koneksi ke Database MySQL

Buka file **config.php**, silakan isi dengan kode berikut.

```
<?php

$server = "localhost";
$user = "root";
$password = "";
$nama_database = "rs_santoso";

$db = mysqli_connect($server, $user, $password, $nama_database);

if( !$db ){
    die("Gagal terhubung dengan database: " . mysqli_connect_error());
}

?>
```

Untuk menghubungkan PHP dengan MySQL, kita menggunakan fungsi **mysqli\_connect()** dengan parameter alamat server, user, password, dan nama database. Silakan isi password sesuai dengan yang dibuat di server-nya. Jika tidak menggunakan password, maka dikosongkan saja. Jika koneksi gagal, variabel `$db` akan bernilai `false`. Karena itu, kita bisa mengeceknya dengan ekspresi `if`.

## Membuat Halaman Indeks

Halaman indeks (`index.php`) pada umumnya merupakan halaman awal dari suatu aplikasi web. Rencananya kita akan menaruh dua link di halaman indeks. Link pertama untuk menuju ke formulir pendaftaran dan link yang kedua untuk menuju ke halaman data pasien yang sudah mendaftar. Gambarnya seperti berikut.

The diagram illustrates the navigation from the main menu to the registration and data pages. The 'Pendaftaran Pasien Baru' menu contains two links: 'Daftar Baru' and 'Data Pasien'. 'Daftar Baru' leads to the 'Formulir Pendaftaran Pasien Baru' page, and 'Data Pasien' leads to the 'Data Pasien' page.

**Pendaftaran Pasien Baru**

**RS Santoso**

**Menu**

- [Daftar Baru](#)
- [Data Pasien](#)

**Formulir Pendaftaran Pasien Baru**

Nama:

Alamat:

Jenis Kelamin: ☐ Laki-laki ☐ Perempuan

Tanggal Lahir:

**Data Pasien**

[\[+\] Tambah Baru](#)

Nama	Alamat	Jenis Kelamin	Tanggal Lahir	Tindakan
River Jusuf	Kebayoran Lama Baru, RT 1 RW 20, Jakarta Tenggara	L	1994-12-31	<a href="#">Edit</a>   <a href="#">Hapus</a>

Total: 1

Sekarang, silakan isi file **index.php** dengan kode berikut:

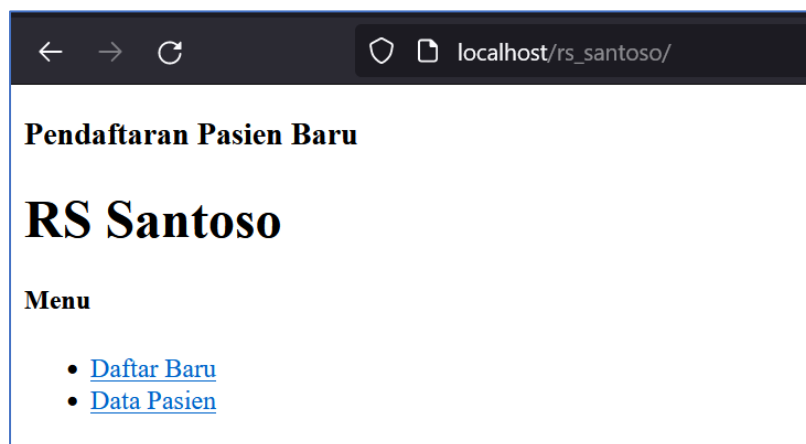
```
<!DOCTYPE html>
<html>
<head>
  <title>Pendaftaran Pasien Baru | RS Santoso</title>
</head>

<body>
  <header>
    <h3>Pendaftaran Pasien Baru</h3>
    <h1>RS Santoso</h1>
  </header>

  <h4>Menu</h4>
  <nav>
    <ul>
      <li><a href="form-daftar.php">Daftar Baru</a></li>
      <li><a href="list-pasien.php">Data Pasien</a></li>
    </ul>
  </nav>

</body>
</html>
```

Ketika mengakses web app kita, server akan langsung mengarahkan ke index.php



## Menampilkan Data dari Tabel di Server MySQL

Silakan buka **list-pasien.php** dan isi dengan kode berikut.

```
<?php include("config.php"); ?>

<!DOCTYPE html>
<html>
<head>
  <title>Pendaftaran Pasien Baru | RS Santoso</title>
</head>

<body>
```

```

<header>
    <h3>Data Pasien</h3>
</header>

<nav>
    <a href="form-daftar.php">[+] Tambah Baru</a>
</nav>

<br>

<table border="1">
<thead>
    <tr>
        <th>Nama</th>
        <th>Alamat</th>
        <th>Jenis Kelamin</th>
        <th>Tanggal Lahir</th>
        <th>Tindakan</th>
    </tr>
</thead>
<tbody>

    <?php
    $sql = "SELECT * FROM pasien";
    $query = mysqli_query($db, $sql);

    while($pasien = mysqli_fetch_array($query)) {
        echo "<tr>";

        echo "<td>".$pasien['nama'].</td>";
        echo "<td>".$pasien['alamat'].</td>";
        echo "<td>".$pasien['jenis_kelamin'].</td>";
        echo "<td>".$pasien['tanggal_lahir'].</td>";

        echo "<td>";
        echo "<a href='form-edit.php?id=".$pasien['id']."'>Edit</a> | ";

        echo "<a href='hapus.php?id=".$pasien['id']."'>Hapus</a>";
        echo "</td>";

        echo "</tr>";
    }
    ?>

</tbody>
</table>

<p>Total: <?php echo mysqli_num_rows($query) ?></p>

</body>
</html>

```

Mari kita bedah kode program tersebut.

Pertama kita membutuhkan koneksi ke database untuk bisa menampilkan data yang tersimpan di database. Maka dari itu kita harus mengimport file **config.php** agar variabel

`$db` dapat dibaca oleh file **list-pasien.php**. Variabel tersebut dibutuhkan saat kita akan menjalankan kueri SQL menggunakan fungsi `mysqli_query()`.

```
<?php include("config.php"); ?>
```

Setelah itu, kita melakukan query untuk mengambil data pasien pada tabel pasien.

```
$sql = "SELECT * FROM pasien";  
$query = mysqli_query($db, $sql);
```

Hasil dari query tersebut, kemudian kita ubah menjadi array dengan fungsi `mysqli_fetch_array()` dan hasilnya disimpan dalam variabel `$`.

```
$pasien = mysqli_fetch_array($query)
```

Perulangan while akan mengulang selama masih ada isi dalam tabel. Kemudian, selama perulangan...kita tampilkan datanya.

```
while($pasien = mysqli_fetch_array($query)){  
    echo "<tr>";  
  
    echo "<td>".$pasien['nama']."</td>";  
    echo "<td>".$pasien['alamat']."</td>";  
    echo "<td>".$pasien['jenis_kelamin']."</td>";  
    echo "<td>".$pasien['tanggal_lahir']."</td>";  
  
    echo "<td>";  
    echo "<a href='form-edit.php?id=".$pasien['id'].">Edit</a> |";  
  
    echo "<a href='hapus.php?id=".$pasien['id'].">Hapus</a>";  
    echo "</td>";  
  
    echo "</tr>";  
}  
?>
```

Perhatikan array `$pasien`, indeksnya sesuai dengan nama kolom pada database dan bersifat *case sensitive*.

Terakhir, kita menampilkan total data yang ada di dalam tabel dengan fungsi `mysqli_num_rows()`.

```
echo mysqli_num_rows($query);
```



Data Pasien				
<a href="#">[+] Tambah Baru</a>				
Nama	Alamat	Jenis Kelamin	Tanggal Lahir	Tindakan
River Jusuf	Kebayoran Lama Baru, RT 1 RW 20, Jakarta Tenggara	L	1994-12-31	<a href="#">Edit</a> <a href="#">Hapus</a>
Total: 1				

## Menambahkan Data ke Tabel di Server MySQL

Ada dua file yang kita butuhkan untuk menambahkan data ke MySQL.

1. **form-daftar.php** halaman untuk form input;
2. **proses-daftar.php** script untuk memproses data yang diinputkan.

Sebenarnya bisa saja dibuat dalam satu file, tapi biar lebih mudah dipahami...kita buat terpisah saja.

Silakan buka file **form-daftar.php** dan isi dengan kode berikut.

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulir Pendaftaran Pasien Baru | RS Santoso</title>
</head>
<body>
  <header>
    <h3>Formulir Pendaftaran Pasien Baru</h3>
  </header>

  <form action="proses-daftar.php" method="POST">

    <fieldset>

      <p>
        <label for="nama">Nama: </label>
        <input type="text" name="nama" placeholder="nama lengkap" />
      </p>
      <p>
        <label for="alamat">Alamat: </label>
        <textarea name="alamat"></textarea>
      </p>
      <p>
        <label for="jenis_kelamin">Jenis Kelamin: </label>
```

```

        <label><input type="radio" name="jenis_kelamin" value="L">
Laki-laki</label>
        <label><input type="radio" name="jenis_kelamin" value="P">
Perempuan</label>
    </p>
    <p>
        <label for="tanggal_lahir">Tanggal Lahir:</label>
        <input type="date" name="tanggal_lahir" placeholder="YYYY-MM-
DD" />
    </p>
    <p>
        <input type="submit" value="Daftar" name="daftar" />
    </p>

    </fieldset>

</form>

</body>
</html>

```

Perhatikan pada tag pembuka `<form>`, di sana kita menggunakan atribut **action** dan **method**.

```

<form action="proses-pendaftaran.php" method="POST">
    . . .

```

Atribut **action** digunakan untuk menentukan script mana yang akan memproses data input, sedangkan atribut **method** untuk menentukan metode pengiriman datanya.

Perhatikan juga pada input-nya, di sana ada atribut **name**. Atribut ini untuk menentukan kunci indeks dari array `$_POST` dan `$_GET`.

The screenshot shows a web browser window with the address bar displaying 'localhost/rs\_santoso/form-daftar.php'. The page title is 'Formulir Pendaftaran Pasien Baru'. The form contains the following elements:

- Nama:** A text input field with the placeholder text 'nama lengkap'.
- Alamat:** A text input field with a placeholder icon.
- Jenis Kelamin:** Two radio buttons labeled 'Laki-laki' and 'Perempuan'.
- Tanggal Lahir:** A date input field with the placeholder 'mm / dd / yyyy' and a calendar icon.
- Daftar:** A button to submit the form.

Selanjutnya isi file **proses-daftar.php** dengan kode berikut.

```
<?php

include("config.php");

// cek apakah tombol daftar sudah diklik atau blum?
if(isset($_POST['daftar'])){

    // ambil data dari formulir
    $nama = $_POST['nama'];
    $alamat = $_POST['alamat'];
    $jk = $_POST['jenis_kelamin'];
    $lahir = $_POST['tanggal_lahir'];

    // buat query
    $sql = "INSERT INTO pasien (nama, alamat, jenis_kelamin, tanggal_lahir)
VALUES ('$nama', '$alamat', '$jk', '$lahir')";
    $query = mysqli_query($db, $sql);

    // apakah query simpan berhasil?
    if( $query ) {
        // kalau berhasil alihkan ke halaman index.php dengan status=sukses
        header('Location: index.php?status=sukses');
    } else {
        // kalau gagal alihkan ke halaman indek.php dengan status=gagal
        header('Location: index.php?status=gagal');
    }

} else {
    die("Akses dilarang...");
}

?>
```

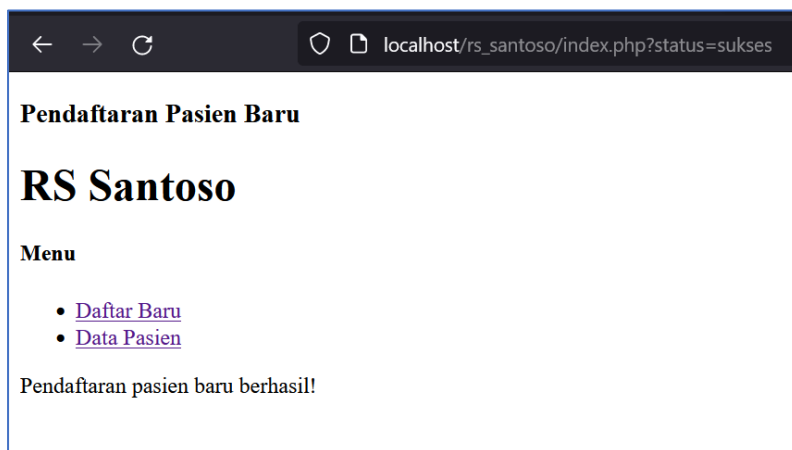
Pertama kita ambil data yang dikirim dari form, semuanya tersimpan pada variabel `$_POST`. Karena tadi kita menggunakan method POST pada form.

Setelah itu, kita buat query penyimpanan.

Lalu melakukan pengecekan, apakah query-nya berhasil atau gagal.

Untuk menampilkan pesan berhasil dan gagal pada halaman `index.php`, silakan tambahkan kode berikut di bawah tag `<nav>` pada file `index.php`.

```
<?php if(isset($_GET['status'])): ?>
    <p>
        <?php
            if($_GET['status'] == 'sukses'){
                echo "Pendaftaran pasien baru berhasil!";
            } else {
                echo "Pendaftaran gagal!";
            }
        ?>
    </p>
<?php endif; ?>
```



**Data Pasien**

[\[+\] Tambah Baru](#)

Nama	Alamat	Jenis Kelamin	Tanggal Lahir	Tindakan
River Jusuf	Kebayoran Lama Baru, RT 1 RW 20, Jakarta Tenggara	L	1994-12-31	<a href="#">Edit</a>   <a href="#">Hapus</a>
Astri	New York City, Purwakarta	P	2024-05-10	<a href="#">Edit</a>   <a href="#">Hapus</a>
Michael Darjon	Desa Paris, Purwakarta	L	2014-03-05	<a href="#">Edit</a>   <a href="#">Hapus</a>

Total: 3

## Menghapus Data dari Tabel di Server MySQL

Kita membutuhkan id untuk menentukan data yang akan dihapus. Karena itu, kita mengirimkan id melalui *query string*.

Contoh:

```
http://localhost/rs_santoso/hapus.php?id=1
```

Query string-nya adalah **?id=1**. Query string akan tersimpan dalam variabel **\$\_GET**.

Untuk lebih jelasnya, silakan buka file **hapus.php** dan isi dengan kode berikut.

```
<?php
include("config.php");
if( isset($_GET['id']) ){
    // ambil id dari query string
    $id = $_GET['id'];

    // buat query hapus
    $sql = "DELETE FROM pasien WHERE id=$id";
    $query = mysqli_query($db, $sql);

    // apakah query hapus berhasil?
    if( $query ){
        header('Location: list-pasien.php');
    } else {
        die("gagal menghapus...");
    }
} else {
    die("akses dilarang...");
}
?>
```

Sekarang coba klik link *hapus*, maka datanya akan langsung terhapus.

**Sebenarnya cara ini kurang aman**, karena kita tidak memberikan dialog konfirmasi kepada user. Misalnya seperti menanyakan ‘Apakah yakin akan menghapus?’

Bagaimana solusinya? 😞

## Update Data di Tabel di Server MySQL

Konsep memperbaharui data hampir sama seperti menghapus. Berikut langkahnya.

1. Ambil **id** data yang akan di-edit.
2. Lakukan query untuk mengambil data dari database berdasarkan id tersebut.
3. Tampilkan data yang telah diperoleh ke dalam form edit data.
4. Setelah data di-edit, jalankan query update untuk menyimpan perubahan.

Untuk lebih jelasnya, silakan buka file **form-edit.php** dan isi dengan kode berikut:

```

<?php

include("config.php");

// kalau tidak ada id di query string
if( !isset($_GET['id']) ){
    header('Location: list-pasien.php');
}

//ambil id dari query string
$id = $_GET['id'];

// buat query untuk ambil data dari database
$sql = "SELECT * FROM pasien WHERE id=$id";
$query = mysqli_query($db, $sql);
$pasien = mysqli_fetch_assoc($query);

// jika data yang di-edit tidak ditemukan
if( mysqli_num_rows($query) < 1 ){
    die("data tidak ditemukan...");
}

?>

<!DOCTYPE html>
<html>
<head>
    <title>Formulir Edit Pasien | RS Santoso</title>
</head>

<body>
    <header>
        <h3>Formulir Edit Pasien</h3>
    </header>

    <form action="proses-edit.php" method="POST">

        <fieldset>

            <input type="hidden" name="id" value="<?php echo $pasien['id']
?>" />

            <p>
                <label for="nama">Nama: </label>
                <input type="text" name="nama" placeholder="nama lengkap"
value="<?php echo $pasien['nama'] ?>" />
            </p>
            <p>
                <label for="alamat">Alamat: </label>
                <textarea name="alamat"><?php echo $pasien['alamat']
?></textarea>
            </p>
            <p>
                <label for="jenis_kelamin">Jenis Kelamin: </label>
                <?php $jk = $pasien['jenis_kelamin']; ?>
                <label><input type="radio" name="jenis_kelamin" value="L" <?php
echo ($jk == 'L') ? "checked": "" ?>> Laki-laki</label>
                <label><input type="radio" name="jenis_kelamin" value="P" <?php
echo ($jk == 'P') ? "checked": "" ?>> Perempuan</label>
            </p>

```

```

        <p>
            <label for="tanggal_lahir">Tanggal Lahir:</label>
            <input type="date" name="tanggal_lahir" placeholder="YYYY-MM-DD" value="<?php echo $pasien['tanggal_lahir'] ?>" />
        </p>

        <p>
            <input type="submit" value="Simpan" name="simpan" />
        </p>

    </fieldset>

</form>

</body>
</html>

```

Ada beberapa hal yang perlu diperhatikan dalam kode tersebut.

Pertama kita mengubah data dari hasil query menjadi array dengan menggunakan fungsi **mysqli\_fetch\_assoc()**. Kenapa menggunakan fungsi tersebut?

Karena data yang akan kita ambil adalah satu baris dan agar indeks array-nya menggunakan nama kolom.

Contoh hasilnya seperti ini:

```

Array
(
    [id] => 3
    [nama] => Michael Darjon
    [alamat] => Desa Paris, Purwakarta
    [jenis_kelamin] => L
    [tanggal_lahir] => 2014-03-05
)

```

Kemudian data tersebut kita tampilkan untuk menjadi nilai default pada form.

Jangan lupa untuk membuat input id dengan tipe *hidden* untuk menyimpan id data yang akan di-update.

```

<input type="hidden" name="id" value="<?php echo $pasien['id'] ?>" />

```

Untuk input radio, kita menggunakan operator ternary agar bisa membuat atribut *checked*.

```

<p>
    <label for="jenis_kelamin">Jenis Kelamin: </label>
    <?php $jk = $pasien['jenis_kelamin']; ?>
    <label><input type="radio" name="jenis_kelamin" value="L" <?php
echo ($jk == 'L') ? "checked": "" ?>> Laki-laki</label>
    <label><input type="radio" name="jenis_kelamin" value="P" <?php
echo ($jk == 'P') ? "checked": "" ?>> Perempuan</label>
</p>

```

Setelah itu, silakan isi file **proses-edit.php** dengan kode berikut.

```

<?php
include("config.php");

// cek apakah tombol simpan sudah diklik atau blum?
if(isset($_POST['simpan'])) {

    // ambil data dari formulir
    $id = $_POST['id'];
    $nama = $_POST['nama'];
    $alamat = $_POST['alamat'];
    $jk = $_POST['jenis_kelamin'];
    $ttl = $_POST['tanggal_lahir'];

    // buat query update
    $sql = "UPDATE pasien SET nama='$nama', alamat='$alamat',
jenis_kelamin='$jk', tanggal_lahir='$ttl' WHERE id=$id";
    $query = mysqli_query($db, $sql);

    // apakah query update berhasil?
    if( $query ) {
        // kalau berhasil alihkan ke halaman list-pasien.php
        header('Location: list-pasien.php');
    } else {
        // kalau gagal tampilkan pesan
        die("Gagal menyimpan perubahan...");
    }

} else {
    die("Akses dilarang...");
}

?>

```



localhost/rs\_santoso/form-edit.php?id=3

### Formulir Edit Pasien

Nama:

Alamat:

Jenis Kelamin: ☒ Laki-laki ☐ Perempuan

Tanggal Lahir:

```

<p>
  <label for="jenis_kelamin">Jenis Kelamin:</label>
  <input type="radio" name="jenis_kelamin" value="L" checked="" />
  Laki-laki
</p>
  <input type="radio" name="jenis_kelamin" value="P" />
  Perempuan
</p>

```

Kita edit alamatnya seperti berikut, kemudian klik simpan.

Alamat:

localhost/rs\_santoso/list-pasien.php

### Data Pasien

[\[+\] Tambah Baru](#)

Nama	Alamat	Jenis Kelamin	Tanggal Lahir	Tindakan
River Jusuf	Kebayoran Lama Baru, RT 1 RW 20, Jakarta Tenggara	L	1994-12-31	<a href="#">Edit</a>   <a href="#">Hapus</a>
Michael Darjon	Desa Paris No.78, Purwakarta	L	2014-03-05	<a href="#">Edit</a>   <a href="#">Hapus</a>

Total: 2

Jika kalian amati, kita sudah berhasil memanfaatkan HTML, PHP, dan juga MySQL untuk membuat web app CRUD sederhana. Akan tetapi, tampilan dari web app-nya masih sangat “primitif”. Untuk itu, kalian bisa memanfaatkan **CSS** untuk mempercantik tampilan antarmuka web yang telah dibuat.

Terima kasih sudah mengikuti tutorial ini sampai akhir.

Selamat belajar! 😊

## References

<https://www.petanikode.com/tutorial-php-mysql/>

[https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)

<https://www.youtube.com/watch?v=v38X2yDigYI>