

UNIVERSITAS GUNADARMA

PRAKTIKUM PEMROGRAMAN WEB



MANUAL BOOK

“Perancangan CRUD Data Klinik Dokter Dengan Menggunakan Bahasa *HTML, PHP*
dan *CSS* Berbasis Website”

Dafa Maulana

50421315

3IA14

Fakultas Teknologi Industri

Teknik Informatika

Ditulis Guna Melengkapi Sebagian Syarat Praktikum Pemrograman WEB

Universitas Gunadarma

2023

BAB I

Pendahuluan

1.1 Latar Belakang

Dalam era digital ini, pemanfaatan teknologi informasi semakin mendominasi berbagai sektor, termasuk dalam bidang kesehatan. Klinik dokter sebagai salah satu layanan kesehatan yang sangat penting pun tidak terlepas dari perkembangan teknologi. Untuk memastikan efisiensi dan keterjangkauan layanan kesehatan, perancangan sistem informasi yang terintegrasi menjadi suatu keharusan.

Salah satu pendekatan yang efektif adalah dengan mengembangkan aplikasi CRUD (Create, Read, Update, Delete) untuk mengelola data klinik dokter. Pemanfaatan bahasa pemrograman web seperti HTML, PHP, dan CSS menjadi pilihan tepat untuk menciptakan antarmuka yang responsif dan mudah digunakan.

Dalam perancangan CRUD data klinik dokter berbasis website, HTML digunakan sebagai dasar struktur halaman web, memberikan kerangka dasar untuk menampilkan informasi. PHP, sebagai bahasa pemrograman server-side, berperan dalam pengolahan data dan interaksi dengan basis data. Sementara itu, CSS digunakan untuk mengatur tata letak dan penampilan halaman, menciptakan antarmuka yang estetis dan mudah dipahami.

Melalui integrasi ketiga bahasa pemrograman ini, sistem ini dapat memfasilitasi pencatatan data dokter dengan cara yang efisien dan terorganisir. Pengguna akan dapat dengan mudah menambah, mengakses, memperbarui, dan menghapus data, meningkatkan produktivitas klinik.

1.2 Tujuan

Tujuan dari perancangan CRUD data klinik dokter berbasis website ini diharapkan dapat memberikan kontribusi positif dalam pengelolaan informasi kesehatan, membantu klinik dokter untuk memberikan layanan yang lebih baik, dan mempercepat proses pengambilan keputusan. Langkah ini sejalan dengan arus transformasi digital di sektor kesehatan, menciptakan pondasi yang kokoh untuk penyediaan layanan kesehatan yang efektif dan terkini.

BAB II

Pembahasan

2.1 HTML

HTML (HyperText Markup Language) adalah sebuah bahasa markup yang digunakan untuk membangun struktur dasar halaman web. Dengan menggunakan serangkaian elemen atau tag, HTML memberikan petunjuk kepada browser web mengenai bagaimana menampilkan teks, gambar, hyperlink, formulir, dan elemen-elemen lainnya. Setiap elemen HTML memiliki fungsi khusus dalam membentuk struktur dokumen, dan hubungan antar elemen tersebut menciptakan tampilan halaman web yang terorganisir dan mudah dimengerti.

Dengan menggunakan HTML, pengembang dapat merancang struktur halaman web dengan jelas, memudahkan interaksi pengguna, dan memberikan dasar untuk integrasi dengan bahasa pemrograman web lainnya, seperti PHP dan JavaScript. Dalam konteks perancangan CRUD data klinik dokter, HTML menjadi fondasi yang kuat untuk membangun antarmuka pengguna yang efisien dan responsif.

2.1.1 Kegunaan HTML

Berikut adalah beberapa kegunaan HTML :

1. Membuat struktur dasar halaman web

HTML digunakan untuk membuat struktur halaman web, seperti header, navbar, body, dan footer. Struktur ini membantu mengorganisasi konten halaman web dan membuat pengalaman pengguna yang lebih baik.

2. Menyusun konten halaman web

HTML digunakan untuk menyusun konten halaman web, seperti paragraf, heading, gambar, link, dan formulir. Tag HTML digunakan untuk mendefinisikan elemen-elemen ini dan bagaimana mereka harus ditampilkan.

3. Menambahkan formulir input data

HTML digunakan untuk menambahkan formulir input data ke halaman web sehingga pengguna dapat memasukkan informasi, seperti nama, email, atau komentar. Formulir ini dibuat dengan menggunakan tag <form> dan berbagai tag input lainnya.

4. Menambahkan konten multimedia

HTML digunakan untuk menambahkan konten multimedia ke halaman web, seperti gambar, video, dan audio. Tag HTML yang berbeda digunakan untuk setiap jenis konten multimedia.

5. Menyiapkan halaman web untuk CSS dan JavaScript

HTML digunakan untuk menyiapkan halaman web untuk CSS dan JavaScript, yang digunakan untuk mengontrol tampilan dan perilaku halaman web. Tag HTML yang berbeda digunakan untuk menyertakan file CSS dan JavaScript.

2.1.2 Kelebihan HTML

Berikut adalah kelebihan HTML :

1. Standar dan didukung oleh semua browser web

HTML adalah bahasa markup standar untuk halaman web, dan didukung oleh semua browser web modern. Ini memastikan bahwa halaman web yang dibuat dengan HTML akan ditampilkan dengan benar di semua browser.

2. Mudah dipelajari dan digunakan

HTML adalah bahasa markup yang relatif mudah dipelajari dan digunakan. Sintaksnya sederhana dan mudah dipahami, dan ada banyak sumber daya yang tersedia untuk membantu pemula belajar HTML.

3. Fleksibel dan ekstensibel

HTML adalah bahasa yang fleksibel dan ekstensibel, dan dapat digunakan untuk membuat berbagai jenis halaman web. Tag HTML baru dapat ditambahkan ke

standar HTML melalui proses yang disebut "standardization", dan ada banyak tag HTML kustom yang tersedia untuk berbagai keperluan.

4. Gratis dan open-source

HTML adalah bahasa gratis dan open-source, sehingga dapat digunakan secara bebas tanpa biaya lisensi. Ini membuatnya menjadi pilihan yang menarik untuk pengembang web yang mencari bahasa markup yang terjangkau dan fleksibel.

2.1.3 Kekurangan HTML

Berikut adalah kekurangan HTML :

1. Tidak memiliki tampilan yang bagus secara default

HTML tidak memiliki tampilan yang bagus secara default, dan memerlukan CSS untuk mengontrol tampilan dan gaya halaman web. Tanpa CSS, halaman web HTML akan terlihat polos dan tidak menarik.

2. Tidak dapat digunakan untuk interaktivitas yang kompleks

HTML tidak dapat digunakan untuk interaktivitas yang kompleks, dan memerlukan JavaScript untuk menambahkan fungsionalitas dinamis ke halaman web. Tanpa JavaScript, halaman web HTML akan menjadi statis dan tidak interaktif.

3. Membutuhkan pemahaman tentang tag dan atribut HTML

Untuk membuat halaman web HTML yang efektif, pengembang web perlu memiliki pemahaman yang baik tentang tag dan atribut HTML. Ada banyak tag dan atribut yang berbeda, dan dapat memakan waktu untuk mempelajari semuanya.

4. Dapat menjadi berulang dan membosankan untuk menulis

Menulis kode HTML dapat menjadi berulang dan membosankan, terutama untuk halaman web yang kompleks. Ada alat dan kerangka kerja yang tersedia untuk membantu pengembang web menulis kode HTML yang lebih cepat dan mudah, tetapi ini masih bisa menjadi hambatan untuk pengembang web pemula.

2.2 PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang digunakan secara luas untuk pengembangan web dinamis. Berbeda dengan HTML yang berperan dalam menentukan struktur dan konten halaman web, PHP berfungsi sebagai bahasa pemrograman server-side yang memproses data di server sebelum hasilnya dikirim ke browser pengguna.

PHP dapat diintegrasikan ke dalam kode HTML untuk membuat halaman web yang dinamis, interaktif, dan responsif. Keunggulan utama PHP adalah kemampuannya untuk berinteraksi dengan database, mengambil informasi dari sana, dan menampilkan hasilnya kepada pengguna dalam format yang diinginkan.

2.2.1 Kegunaan PHP

Berikut adalah beberapa kegunaan utama PHP:

1. Mengelola data

PHP dapat digunakan untuk mengelola data dari database, seperti menambahkan, memperbarui, dan menghapus data. PHP juga dapat digunakan untuk menghasilkan laporan dan grafik dari data.

2. Menghasilkan konten

PHP dapat digunakan untuk menghasilkan konten dinamis, seperti mengubah tampilan halaman web berdasarkan data pengguna atau menghasilkan konten baru berdasarkan data dari database.

3. Berinteraksi dengan database

PHP dapat digunakan untuk berinteraksi dengan database, seperti membuat koneksi ke database, menjalankan kueri database, dan membaca dan menulis data dari database.

4. Membangun aplikasi web

PHP dapat digunakan untuk membangun aplikasi web, seperti aplikasi e-commerce, aplikasi manajemen konten, dan aplikasi web lainnya.

2.2.2 Kelebihan PHP

Berikut adalah kelebihan PHP :

1. Populer

PHP adalah bahasa pemrograman yang paling populer di dunia untuk pengembangan web. Hal ini menjadikannya bahasa yang mudah untuk dipelajari dan diimplementasikan.

2. Mudah dipelajari

PHP memiliki sintaks yang sederhana dan mudah dipahami. Bahasa ini juga memiliki banyak dokumentasi dan tutorial yang tersedia secara online.

3. Dukungan komunitas yang kuat

PHP memiliki komunitas pengguna yang besar dan aktif. Komunitas ini menyediakan dukungan dan bantuan bagi para pengembang PHP di seluruh dunia.

4. Terintegrasi dengan database

PHP memiliki dukungan yang kuat untuk database. Bahasa ini dapat digunakan untuk mengakses dan mengelola data dari berbagai jenis database, seperti MySQL, PostgreSQL, dan SQLite.

5. Multiplatform

PHP dapat dijalankan di berbagai sistem operasi, termasuk Windows, Linux, dan macOS. Hal ini menjadikannya bahasa yang fleksibel dan dapat digunakan di berbagai lingkungan.

2.2.3 Kekurangan PHP

Berikut adalah kekurangan PHP :

1. Keamanan

PHP terkenal dengan masalah keamanannya. Bahasa ini sering digunakan untuk membuat situs web yang rentan terhadap serangan siber. Hal ini disebabkan oleh beberapa faktor, seperti kurangnya standar pengkodean yang ketat, banyaknya library pihak ketiga yang tidak aman, dan kebiasaan pengembang yang kurang memperhatikan masalah keamanan.

2. Performa

PHP adalah bahasa pemrograman yang diinterpretasi, yang berarti bahwa kode PHP tidak perlu dikompilasi sebelum dapat dijalankan. Hal ini dapat menyebabkan masalah performa, terutama untuk aplikasi web yang membutuhkan kinerja tinggi.

3. Sintaks yang tidak konsisten

PHP memiliki sintaks yang tidak konsisten, yang dapat membuat kode PHP menjadi sulit untuk dibaca dan dipahami. Hal ini disebabkan oleh fakta bahwa PHP telah berkembang secara bertahap selama bertahun-tahun, dan tidak ada upaya untuk menstandarkan sintaksnya.

4. Kurang dukungan untuk pengembangan aplikasi besar

PHP tidak terlalu cocok untuk pengembangan aplikasi web yang sangat besar. Hal ini disebabkan oleh beberapa faktor, seperti sintaks yang tidak konsisten, kurangnya dukungan untuk pemrograman modular, dan kurangnya dukungan untuk pengujian otomatis.

5. Kurangnya dukungan untuk package

PHP tidak memiliki dukungan yang kuat untuk package, yang dapat membuat pengembangan aplikasi web menjadi lebih sulit. Hal ini disebabkan oleh fakta bahwa PHP tidak memiliki sistem package bawaan, dan pengembang harus menggunakan tool pihak ketiga untuk mengelola package mereka.

2.3 CSS

CSS (Cascading Style Sheets) adalah bahasa pemrograman yang digunakan untuk mengatur tampilan dan gaya dari halaman web yang menggunakan markup language seperti HTML atau XML. Fungsinya adalah untuk memisahkan struktur konten dari desain visual, memungkinkan pengguna untuk mengontrol bagaimana elemen-elemen pada halaman web tersebut ditampilkan pada layar, di cetak, atau di media lainnya.

CSS memungkinkan pengguna untuk menentukan berbagai properti gaya seperti warna, ukuran teks, jenis font, layout, dan tata letak halaman. Penggunaan CSS yang efektif dapat membuat halaman web menjadi lebih responsif, cepat dimuat, dan mudah diakses melalui perangkat yang berbeda-beda seperti komputer desktop, tablet, atau ponsel. Selain itu, dengan konsep cascading-nya, CSS memungkinkan pengguna untuk menimpa aturan gaya yang ada, sehingga memudahkan pengembangan dan pemeliharaan situs web dengan skala yang beragam.

2.3.1 Kegunaan CSS

Berikut adalah beberapa kegunaan CSS:

1. Mengatur gaya teks

Dengan CSS, Anda dapat mengatur gaya teks seperti ukuran font, jenis font, warna font, tebal font, miring font, dan masih banyak lagi.

2. Mengatur warna latar belakang

CSS dapat digunakan untuk mengatur warna latar belakang halaman web, elemen HTML tertentu, atau bahkan bagian tertentu dari elemen HTML.

3. Mengatur jarak dan margin

CSS dapat digunakan untuk mengatur jarak antara elemen HTML, margin elemen HTML dari tepi halaman, dan padding elemen HTML dari dalam elemen HTML.

4. Mengatur posisi elemen HTML

CSS dapat digunakan untuk mengatur posisi elemen HTML pada halaman web, seperti posisi absolut, posisi tetap, dan posisi relatif.

5. Mengatur tampilan elemen HTML

CSS dapat digunakan untuk mengatur tampilan elemen HTML, seperti menyembunyikan elemen HTML, mengubah transparansi elemen HTML, dan mengubah tampilan elemen HTML menjadi elemen lain.

6. Membuat animasi

CSS dapat digunakan untuk membuat animasi pada elemen HTML, seperti animasi transisi elemen HTML, animasi transformasi elemen HTML, dan animasi keyframes.

7. Menyesuaikan tampilan halaman web untuk perangkat yang berbeda

CSS dapat digunakan untuk menyesuaikan tampilan halaman web untuk perangkat yang berbeda, seperti komputer desktop, laptop, tablet, dan smartphone.

8. Membuat tema untuk halaman web

CSS dapat digunakan untuk membuat tema untuk halaman web, sehingga halaman web memiliki tampilan yang konsisten dan menarik.

9. Meningkatkan kecepatan loading halaman web

Dengan menggunakan CSS dengan benar, Anda dapat meningkatkan kecepatan loading halaman web, sehingga halaman web Anda dapat dimuat lebih cepat oleh pengguna.

10. Membuat halaman web yang dapat diakses

CSS dapat digunakan untuk membuat halaman web yang dapat diakses oleh pengguna dengan disabilitas, seperti pengguna dengan gangguan penglihatan.

2.3.2 Kelebihan CSS

Berikut adalah kelebihan CSS :

1. Fleksibilitas

CSS memungkinkan pengembang untuk mengatur tampilan halaman web dengan cara yang fleksibel dan mudah disesuaikan.

2. Efisiensi

CSS dapat digunakan untuk mengatur tampilan elemen HTML secara berulang-ulang, sehingga dapat menghemat waktu dan tenaga pengembang.

3. Kekuatan

CSS dapat digunakan untuk membuat berbagai efek visual yang kompleks dan menarik.

4. Keterjangkauan

CSS adalah bahasa pemrograman yang gratis dan dapat digunakan oleh siapa saja.

2.3.3 Kekurangan CSS

Berikut adalah kekurangan CSS :

1. Kompleksitas

CSS dapat menjadi bahasa yang kompleks untuk dipelajari, terutama untuk pengembang yang baru memulai.

2. Kesalahan

Kesalahan dalam penulisan kode CSS dapat menyebabkan tampilan halaman web menjadi tidak sesuai dengan yang diinginkan.

3. Kompatibilitas

CSS harus kompatibel dengan berbagai browser web, sehingga pengembang harus memastikan bahwa kode CSS mereka dapat berjalan dengan baik di berbagai browser.

2.4 Hubungan antara HTML, PHP dan CSS

HTML (HyperText Markup Language), PHP (Hypertext Preprocessor), dan CSS (Cascading Style Sheets) merupakan tiga teknologi utama yang sering digunakan bersama-sama untuk membangun halaman web dinamis dan menarik.

Berikut adalah kesimpulan tentang hubungan antara ketiganya :

1. HTML digunakan untuk membuat struktur dasar dan konten halaman web.
2. CSS digunakan untuk mengatur tata letak dan gaya visual dari elemen-elemen HTML yang dibuat dengan HTML.
3. PHP digunakan untuk mengelola logika dan data dinamis, yang kemudian dapat disisipkan ke dalam halaman HTML.
4. Dengan bahasa pemrograman HTML, CSS, dan PHP memungkinkan pembuatan halaman web yang dinamis, interaktif, dan estetis.

Dengan mengintegrasikan ketiganya, pengembang dapat menciptakan pengalaman pengguna yang lebih kaya, dengan memisahkan struktur, tata letak, dan logika aplikasi, sehingga memudahkan pemeliharaan dan pengembangan lebih lanjut.

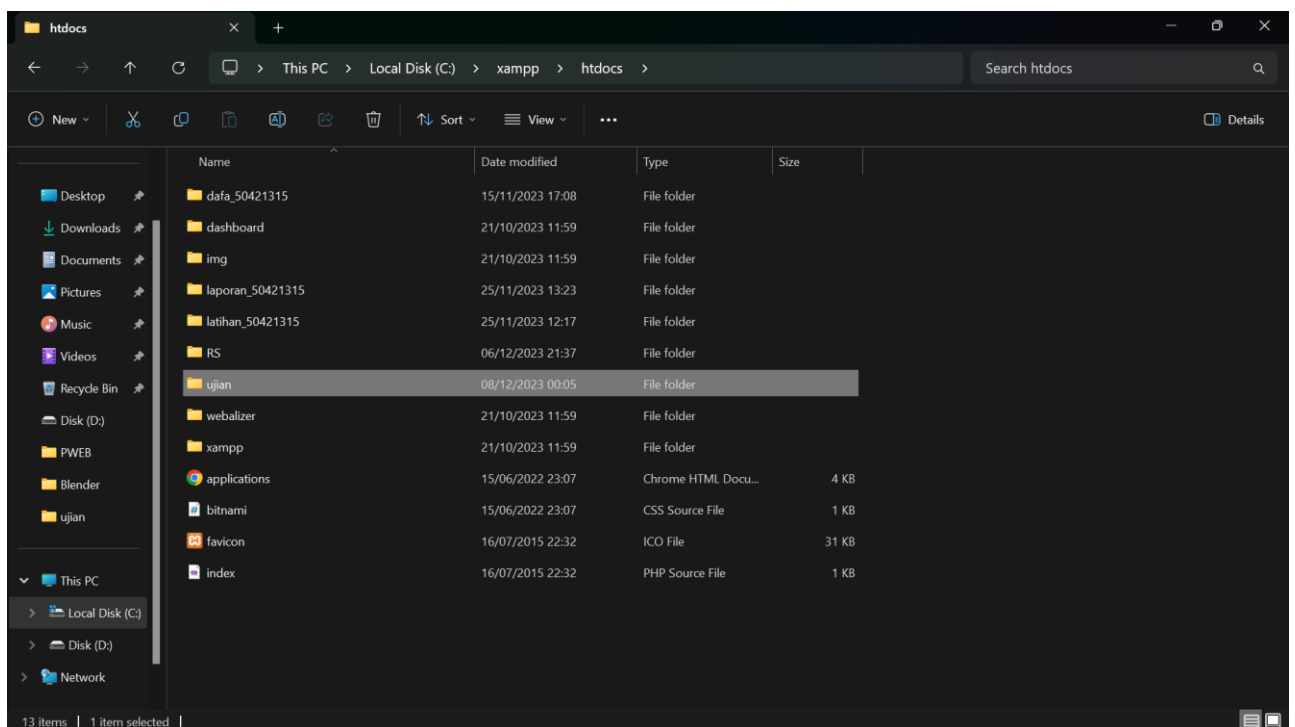
BAB III

Analisa & Perancangan

Projek yang saya buat untuk ujian yaitu, Perancangan CRUD Data Klinik Dokter dengan menggunakan Bahasa pemrograman *HTML, PHP dan CSS* berbasis website dengan memanfaatkan Visual Studio Code dan PHPmyadmin sebagai tempat pengaplikasiannya.

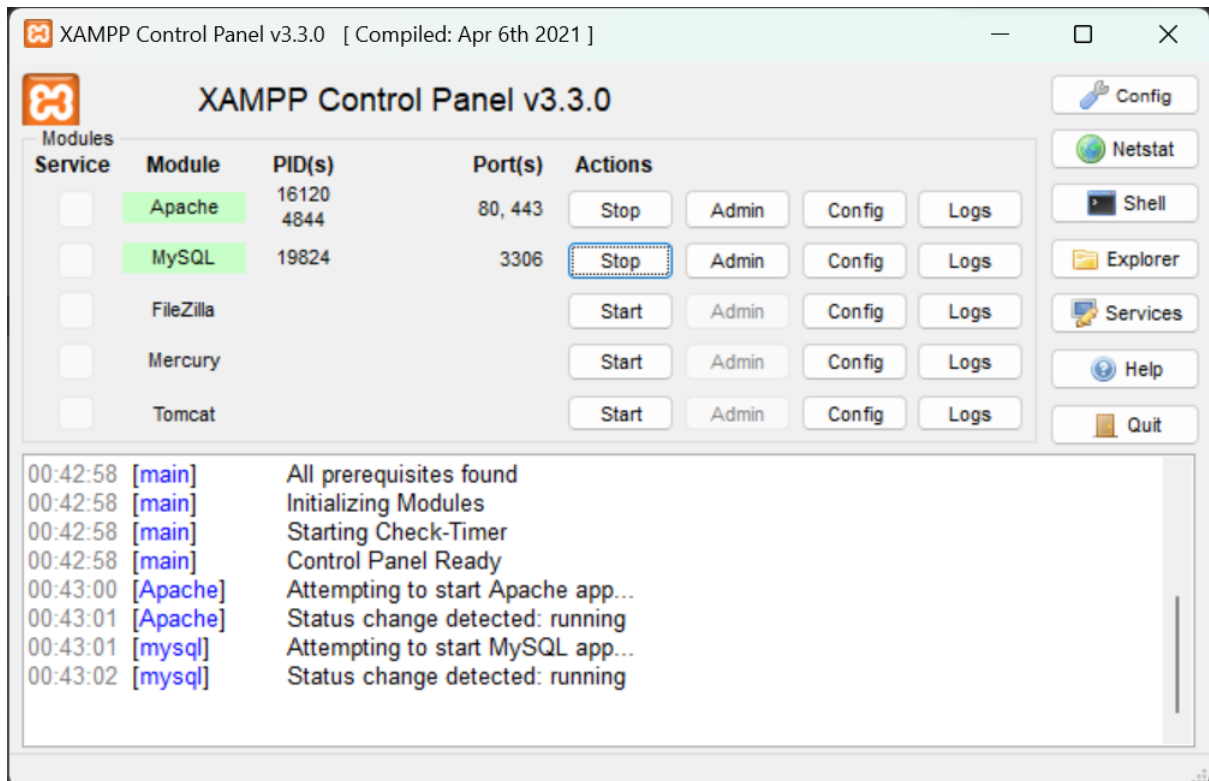
Berikut Langkah-langkah pembuatan Database yang digunakan dalam projek ini :

1. Membuat folder pada local disk C: xampp/htdocs (Folder saya bernama “ujian”).



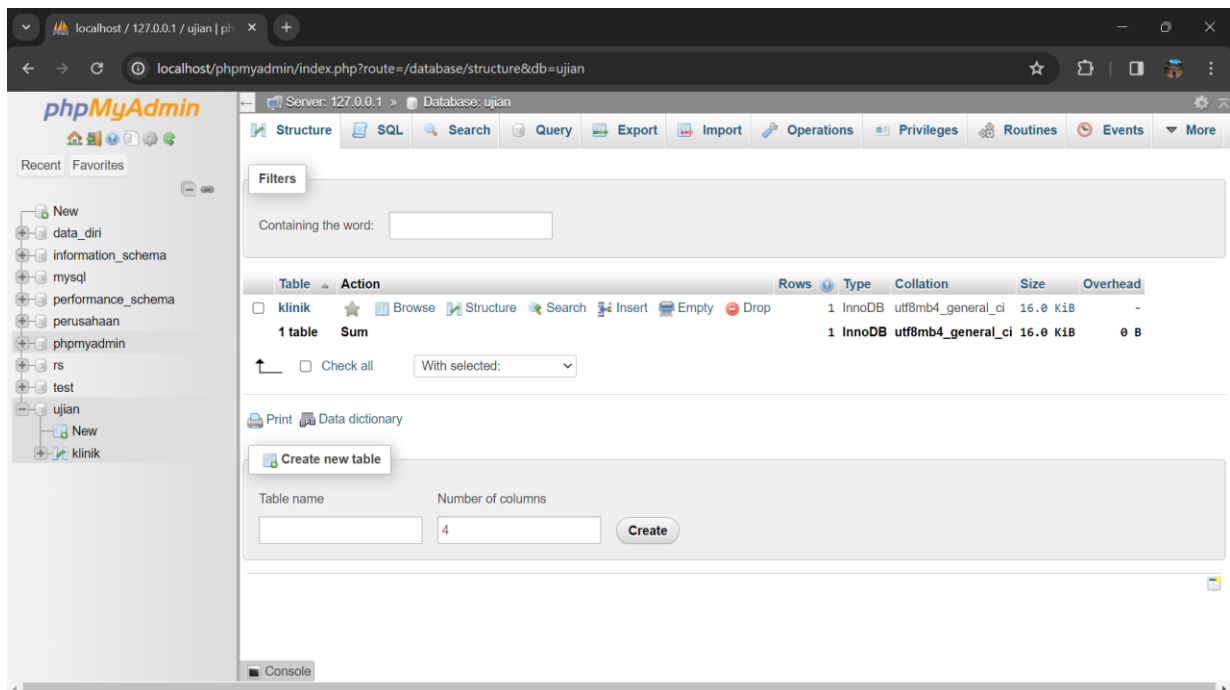
Gambar 3.1 Membuat Folder

2. Membuka aplikasi xampp untuk membuat database.



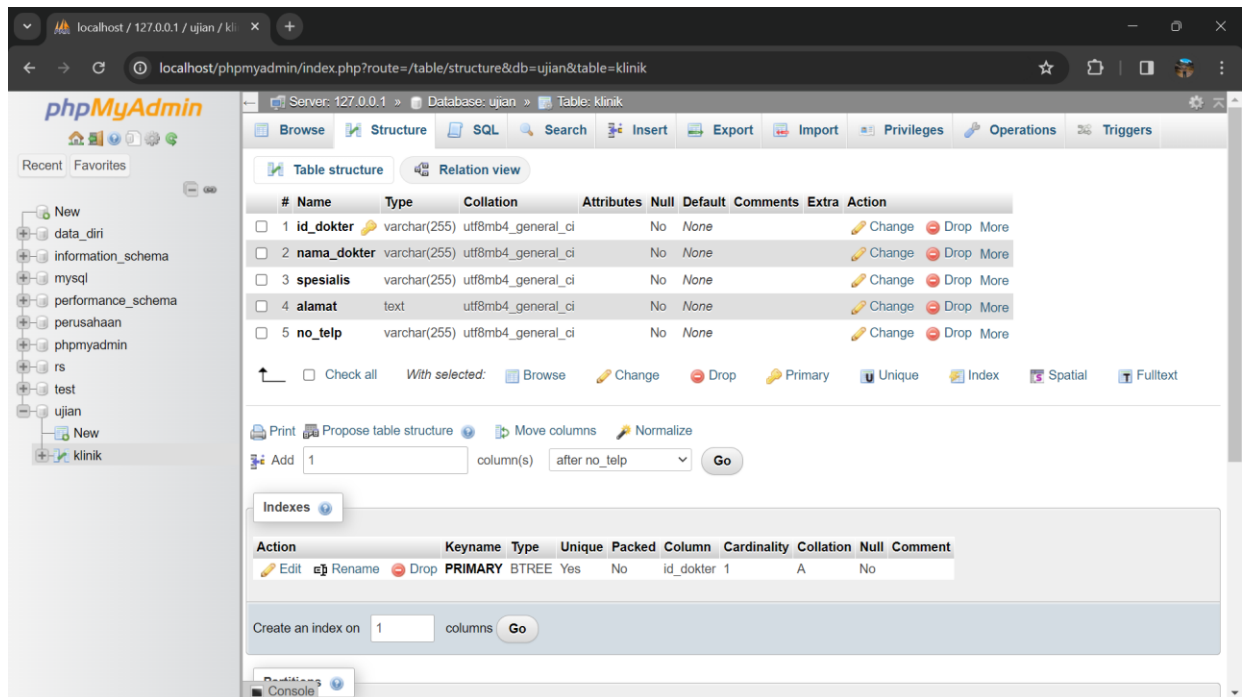
Gambar 3.2 Aplikasi Xampp

3. Membuat database pada phpmyadmin bernama ujian.



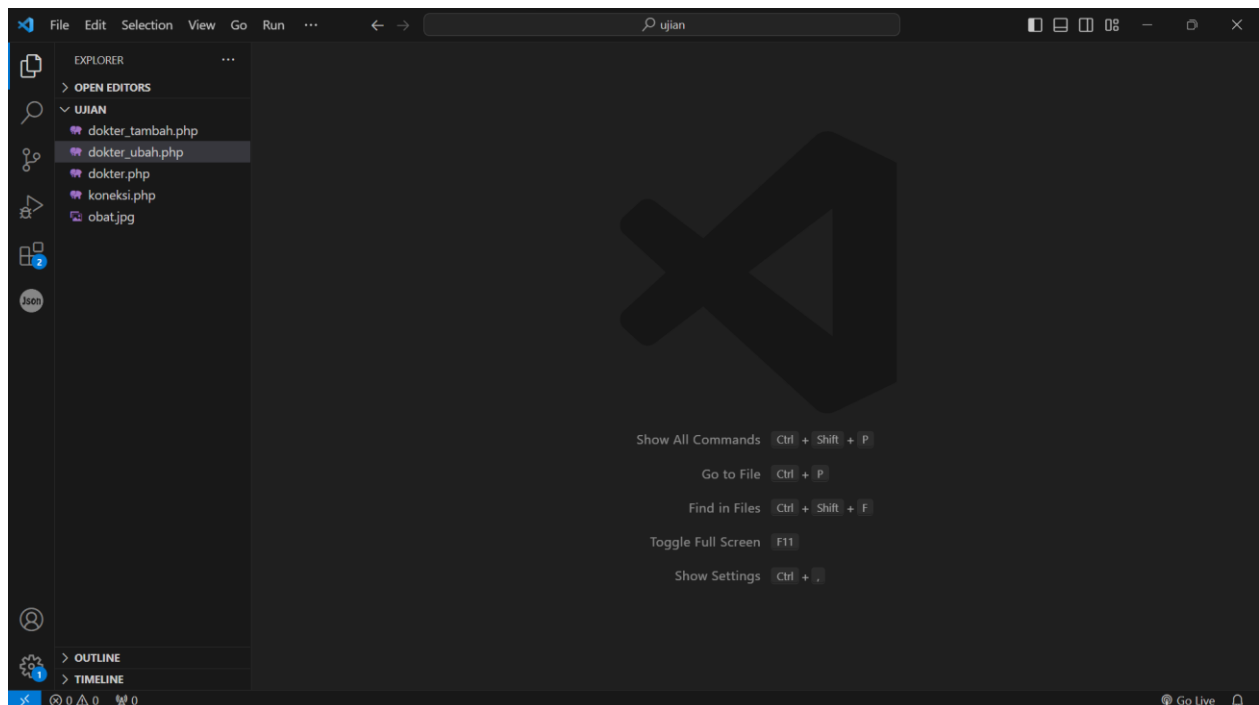
Gambar 3.3 Database ujian

4. Membuat table pada database ujian bernama “Klinik”.

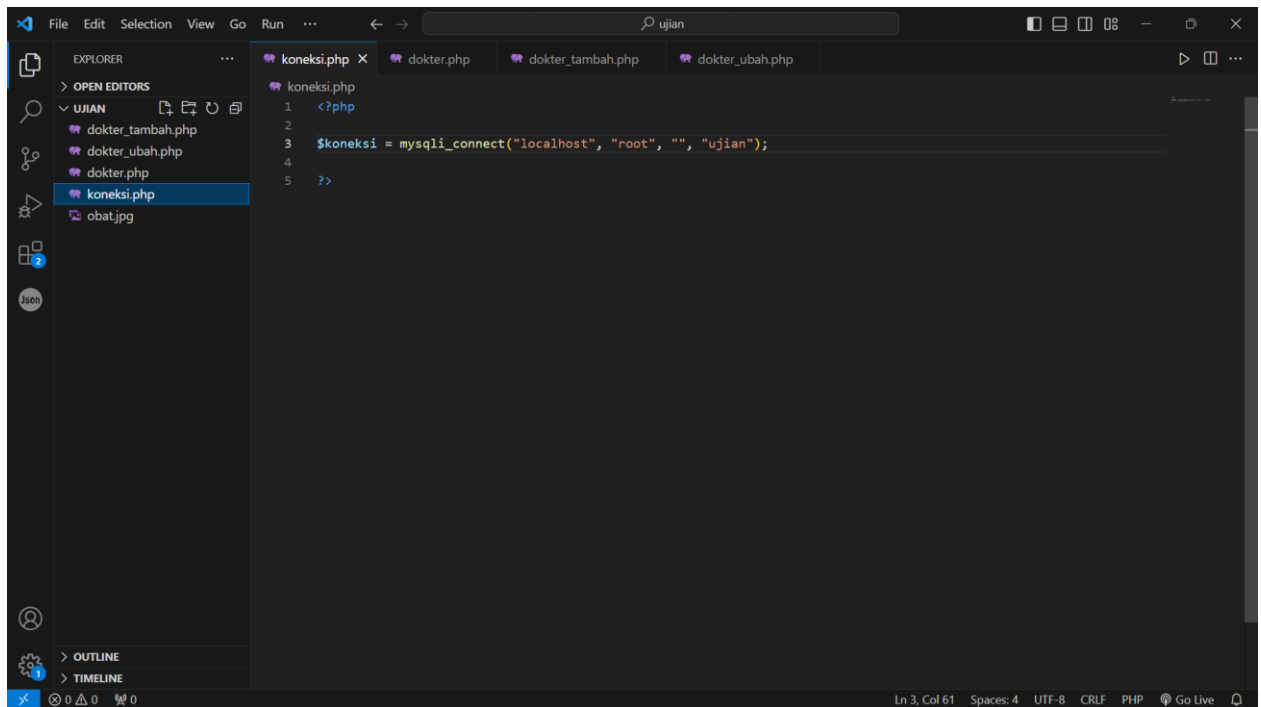


Gambar 3.4 Table data klinik

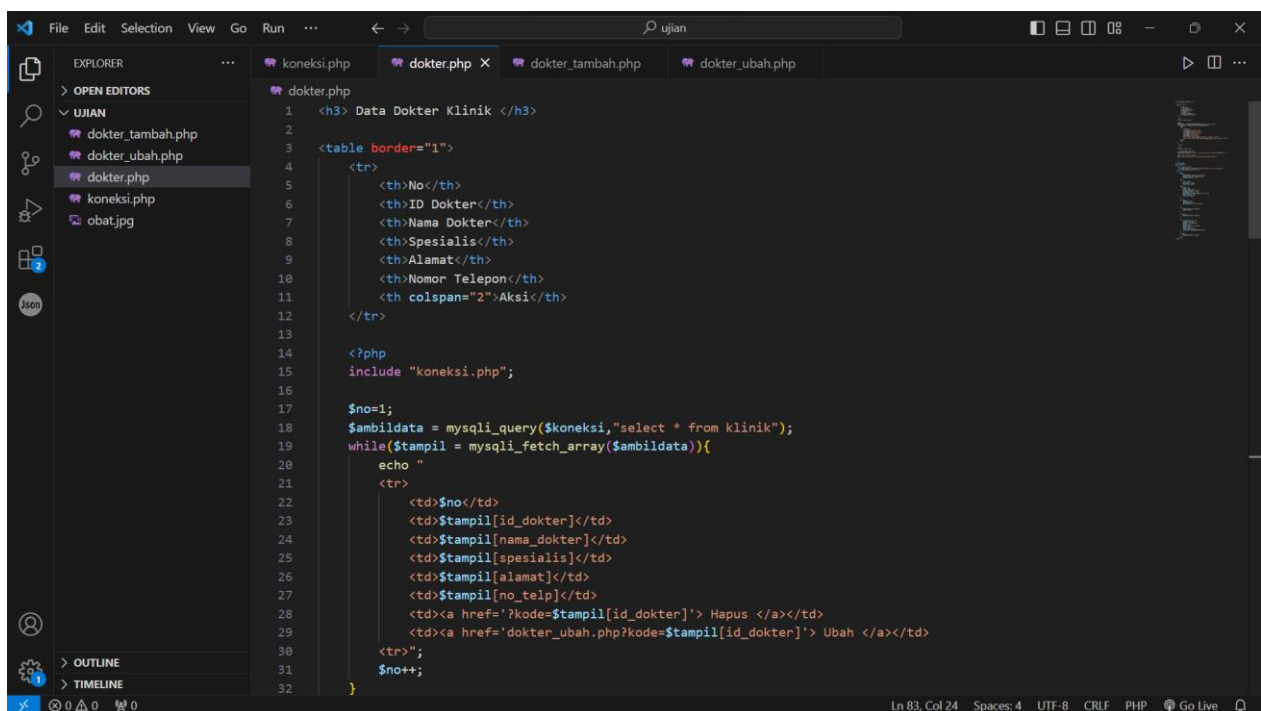
5. Buka folder “ujian” melalui Visual Studio Code, lalu buat file “koneksi.php”, “dokter.php”, “dokter_tambah.php”, “dokter_ubah.php”.



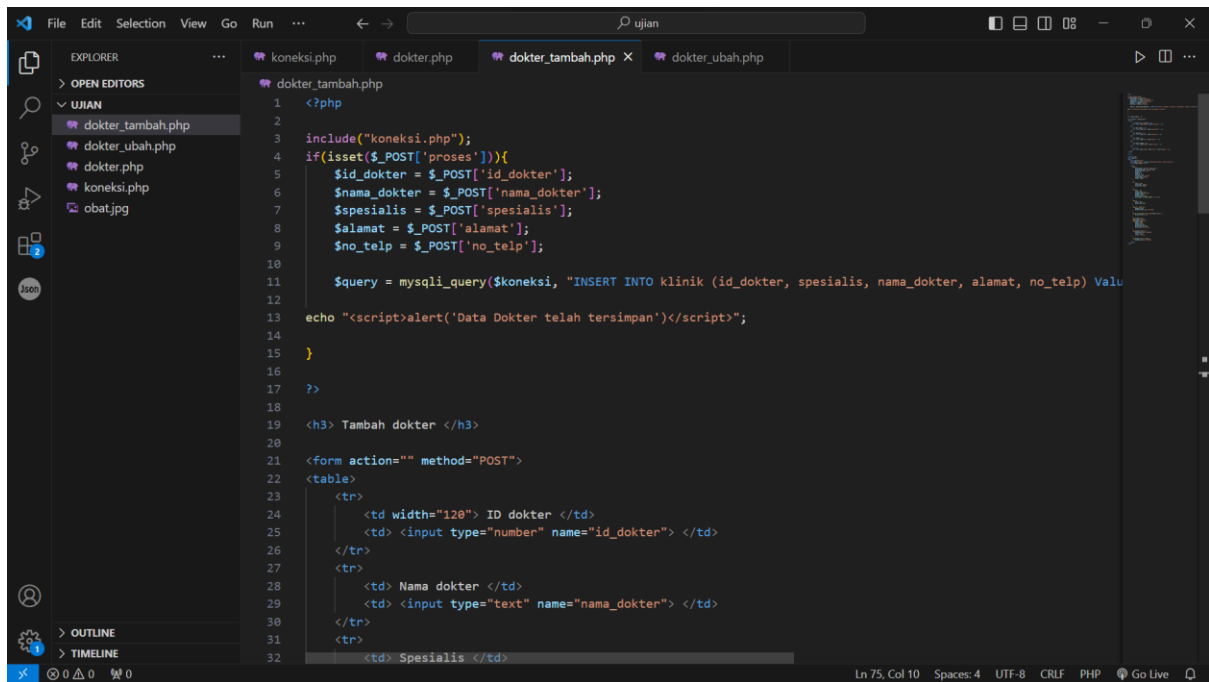
Gambar 3.5 Visual Studio Code



Gambar 3.6 koneksi.php

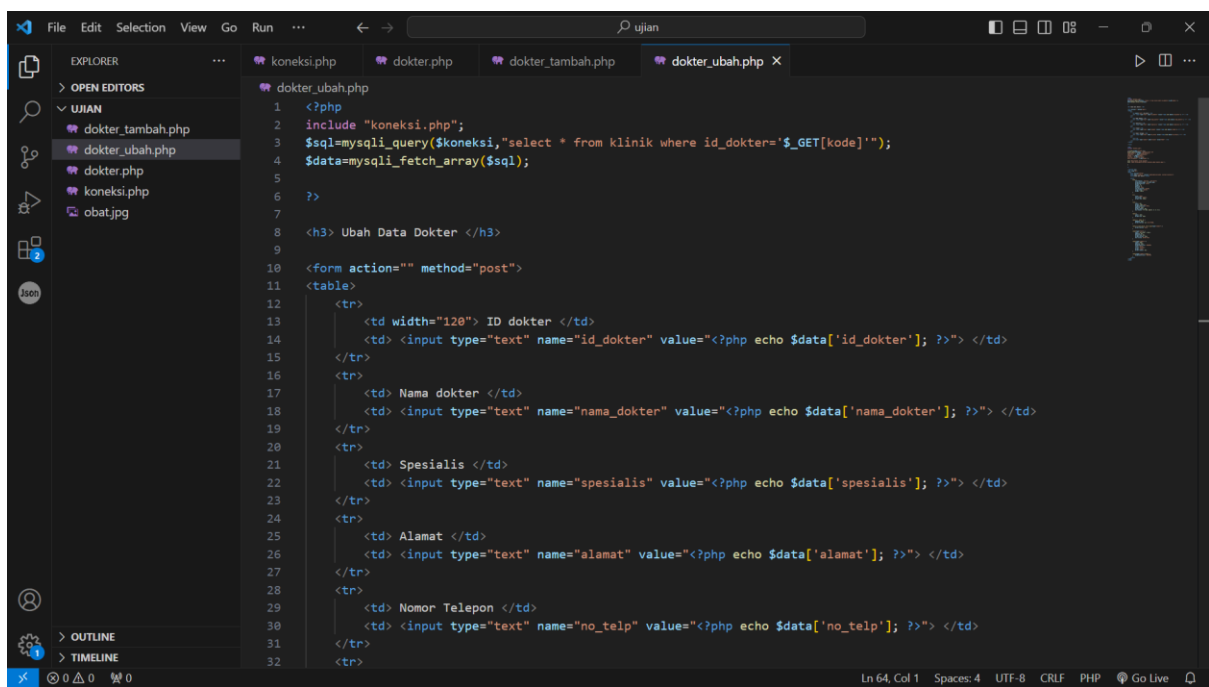


Gambar 3.7 dokter.php



```
1 <?php
2
3 include("koneksi.php");
4 if(isset($_POST['proses'])){
5     $id_dokter = $_POST['id_dokter'];
6     $nama_dokter = $_POST['nama_dokter'];
7     $spesialis = $_POST['spesialis'];
8     $alamat = $_POST['alamat'];
9     $no_telp = $_POST['no_telp'];
10
11     $query = mysqli_query($koneksi, "INSERT INTO klinik (id_dokter, spesialis, nama_dokter, alamat, no_telp) Values ($id_dokter, '$spesialis', '$nama_dokter', '$alamat', '$no_telp')");
12
13     echo "<script>alert('Data Dokter telah tersimpan')</script>";
14 }
15
16 ?>
17
18 <h3> Tambah dokter </h3>
19
20 <form action="" method="POST">
21 <table>
22 <tr>
23 <td width="120"> ID dokter </td>
24 <td> <input type="number" name="id_dokter"> </td>
25 </tr>
26 <tr>
27 <td> Nama dokter </td>
28 <td> <input type="text" name="nama_dokter"> </td>
29 </tr>
30 <tr>
31 <td> Spesialis </td>
```

Gambar 3.8 dokter_tambah.php



```
1 <?php
2 include "koneksi.php";
3 $sql=mysqli_query($koneksi,"select * from klinik where id_dokter='".$_GET[kode']."'");
4 $data=mysqli_fetch_array($sql);
5
6 ?>
7
8 <h3> Ubah Data Dokter </h3>
9
10 <form action="" method="post">
11 <table>
12 <tr>
13 <td width="120"> ID dokter </td>
14 <td> <input type="text" name="id_dokter" value="<?php echo $data['id_dokter']; ?>"> </td>
15 </tr>
16 <tr>
17 <td> Nama dokter </td>
18 <td> <input type="text" name="nama_dokter" value="<?php echo $data['nama_dokter']; ?>"> </td>
19 </tr>
20 <tr>
21 <td> Spesialis </td>
22 <td> <input type="text" name="spesialis" value="<?php echo $data['spesialis']; ?>"> </td>
23 </tr>
24 <tr>
25 <td> Alamat </td>
26 <td> <input type="text" name="alamat" value="<?php echo $data['alamat']; ?>"> </td>
27 </tr>
28 <tr>
29 <td> Nomor Telepon </td>
30 <td> <input type="text" name="no_telp" value="<?php echo $data['no_telp']; ?>"> </td>
31 </tr>
32 <tr>
```

Gambar 3.9 dokter_ubah.php

Source Code :

1. koneksi.php



```
1  <?php
2
3  $koneksi = mysqli_connect("localhost", "root", "", "ujian");
4
5  ?>
```

Penjelasan code :

- **Deklarasi Variabel koneksi**

- <?php :

Ini adalah tag pembuka untuk memulai blok kode PHP. Semua kode PHP harus diapit oleh tag ini

- \$koneksi = mysqli_connect("localhost", "root", "", "ujian"); :

- Mendefinisikan variabel \$koneksi yang akan digunakan untuk menyimpan koneksi ke database.
- mysqli_connect adalah fungsi bawaan PHP yang digunakan untuk melakukan koneksi ke server database MySQL.
- Parameter pertama ("localhost") adalah nama server database. Dalam hal ini, server database berada di localhost atau mesin yang sama dengan script PHP.
- Parameter kedua ("root") adalah nama pengguna (username) yang digunakan untuk mengakses database MySQL. Biasanya, "root" adalah pengguna bawaan dengan hak akses penuh.
- Parameter ketiga ("") adalah sandi (password) untuk pengguna database. Dalam contoh ini, tidak ada sandi (kosong).
- Parameter keempat ("ujian") adalah nama database yang akan digunakan

- ?> :

Ini adalah tag penutup untuk menandai akhir dari blok kode PHP.

2. dokter.php

```
1 <h3> Data Dokter Klinik </h3>
2
3 <table border="1">
4   <tr>
5     <th>No</th>
6     <th>ID Dokter</th>
7     <th>Nama Dokter</th>
8     <th>Spesialis</th>
9     <th>Alamat</th>
10    <th>Nomor Telepon</th>
11    <th colspan="2">Aksi</th>
12  </tr>
13
14  <?php
15    include "koneksi.php";
16
17    $no=1;
18    $sambildata = mysql_query($koneksi,"select * from klinik");
19    while($stampil = mysql_fetch_array($sambildata)){
20      echo "
21        <tr>
22          <td>$no</td>
23          <td>$stampil[id_dokter]</td>
24          <td>$stampil[nama_dokter]</td>
25          <td>$stampil[spesialis]</td>
26          <td>$stampil[alamat]</td>
27          <td>$stampil[no_telp]</td>
28          <td><a href=?kode=$stampil[id_dokter]> Hapus </a></td>
29          <td><a href=?kode=$stampil[id_dokter]> Ubah </a></td>
30        </tr>";
31      $no++;
32    }
33  ?>
34 </table>
35
36  <?php
37    include "koneksi.php";
38
39    if(isset($_GET['kode'])){
40      mysql_query($koneksi,"delete from klinik where id_dokter='$_GET[kode]'");
41
42      echo "Data berhasil dihapus";
43      echo "<meta http-equiv=refresh content=2;URL='dokter.php'>";
44
45    }
46  ?>
47
48 <!DOCTYPE html>
49 <html lang="en">
50 <head>
51   <meta charset="UTF-8">
52   <meta name="viewport" content="width=device-width, initial-scale=1.0">
53   <title>Data Dokter Klinik</title>
54 </head>
55 <style>
56   body{
57     font-family: 'Courier', sans-serif;
58     background-image: url(obat.jpg);
59     background-size: cover;
60   }
61   h3 {
62     color: #333;
63     text-align: center;
64     margin-top: 100px;
65   }
66   table {
67     width: 80%;
68     margin: 20px auto;
69     border-collapse: collapse;
70     background-color: #fff;
71     box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
72     border-radius: 8px;
73     overflow: hidden;
74   }
75
76   th, td {
77     padding: 12px;
78     text-align: left;
79     border-bottom: 1px solid #ddd;
80   }
81
82   th {
83     background-color: #4CAF50;
84     color: white;
85   }
86
87   tr:hover {
88     background-color: #f5f5f5;
89   }
90
91   a {
92     text-decoration: none;
93     padding: 8px 16px;
94     display: inline-block;
95     margin: 4px;
96     background-color: #4CAF50;
97     color: white;
98     border-radius: 4px;
99     transition: background-color 0.3s;
100  }
101
102  a:hover {
103    background-color: #45a049;
104  }
105 </style>
106 </html>
```

Penjelasan code :

- **Bagian PHP**

- `include "koneksi.php";`: Menggunakan pernyataan `include` untuk menyertakan file `koneksi.php` yang kemungkinan berisi koneksi ke database MySQL.
- `<?php ... ?>`: Ini adalah blok PHP yang digunakan untuk memproses data dari database dan menampilkan dalam bentuk tabel.
- `$ambildata = mysqli_query($koneksi,"select * from klinik");`: Menjalankan query SQL untuk mengambil semua data dari tabel "klinik".
- `while($tampil = mysqli_fetch_array($ambildata)){ ... }`: Menggunakan perulangan `while` untuk menampilkan setiap baris data sebagai baris tabel HTML.

Bagian penghapusan data:

- Menggunakan `if(isset($_GET['kode'])) { ... }` untuk memeriksa apakah ada parameter "kode" yang diterima melalui URL.
- Jika ada, melakukan penghapusan data dokter berdasarkan ID Dokter yang diterima melalui parameter "kode".
- Setelah penghapusan, memberikan pesan "Data berhasil dihapus" dan melakukan pengalihan (refresh) ke halaman "dokter.php" setelah 2 detik.

- **Bagian HTML dan CSS**

- `<h3> Data Dokter Klinik </h3>`: Ini adalah elemen HTML untuk menampilkan judul "Data Dokter Klinik" dengan gaya teks yang sesuai.
- `<table border="1">...</table>`: Ini adalah elemen HTML untuk membuat tabel. Setiap baris tabel (elemen `<tr>`) memiliki sel (elemen `<td>`) untuk setiap kolom data. Beberapa kolom memiliki aksi tambahan (hapus dan ubah) yang dihubungkan dengan tautan.
- `<style>`: Ini adalah blok gaya CSS untuk mengatur tata letak dan penampilan elemen-elemen HTML. Misalnya, mengatur font, warna, dan tata letak tabel.

3. dokter_tambah.php

```
1 <?php
2
3 include("koneksi.php");
4 if(isset($_POST['proses'])){
5     $id_dokter = $_POST['id_dokter'];
6     $nama_dokter = $_POST['nama_dokter'];
7     $spesialis = $_POST['spesialis'];
8     $alamat = $_POST['alamat'];
9     $no_telp = $_POST['no_telp'];
10
11     $query = mysqli_query($koneksi, "INSERT INTO klinik (id_dokter, spesialis, nama_dokter, alamat, no_telp) Values ('$id_dokter', '$spesialis', '$nama_dokter', '$alamat', '$no_telp')");
12
13     echo "<script>alert('Data Dokter telah tersimpan')</script>";
14
15 }
16
17 ?>
18
19 <h3> Tambah dokter </h3>
20
21 <form action="" method="POST">
22 <table>
23 <tr>
24 <td width="120"> ID dokter </td>
25 <td> <input type="number" name="id_dokter"> </td>
26 </tr>
27 <tr>
28 <td> Nama dokter </td>
29 <td> <input type="text" name="nama_dokter"> </td>
30 </tr>
31 <tr>
32 <td> Spesialis </td>
33 <td> <input type="text" name="spesialis"> </td>
34 </tr>
35 <tr>
36 <td> Alamat </td>
37 <td> <input type="text" name="alamat"> </td>
38 </tr>
39 <tr>
40 <td> Nomor Telepon </td>
41 <td> <input type="number" name="no_telp"> </td>
42 </tr>
43 <tr>
44 <td></td>
45 <td><input type="submit" name="proses" value="Simpan"> </td>
46 </tr>
47 </table>
48 </form>
49
50
51 <!DOCTYPE html>
52 <html lang="en">
53 <head>
54 <meta charset="UTF-8">
55 <meta name="viewport" content="width=device-width, initial-scale=1.0">
56 <title>Tambah Dokter</title>
57
58 <style>
59 body {
60     font-family: 'Courier', sans-serif;
61     background-image: url(obat.jpg);
62     background-size: cover;
63     margin: 0;
64     padding: 0;
65     display: flex;
66     flex-direction: column;
67     align-items: center;
68     height: 100vh;
69 }
70
71 h3 {
72     color: #333;
73     text-align: center;
74     margin-top: 100px;
75 }
76
77 form {
78     width: 50%;
79     margin: 20px auto;
80     background-color: #fff;
81     padding: 20px;
82     border-radius: 8px;
83     box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
84 }
85
86 table {
87     width: 100%;
88     margin-top: 10px;
89 }
90
91 table tr, table td {
92     padding: 10px;
93     border-bottom: 1px solid #ddd;
94 }
95
96 table tr:last-child, form input[type="submit"] {
97     border-bottom: none;
98 }
99
100 input[type="text"],
101 input[type="number"],
102 input[type="submit"] {
103     width: 100%;
104     padding: 8px;
105     margin-top: 60px;
106     margin-bottom: 12px;
107     box-sizing: border-box;
108 }
109
110 input[type="submit"] {
111     background-color: #45a049;
112     color: white;
113     cursor: pointer;
114 }
115
116 input[type="submit"]:hover {
117     background-color: #4CAF50;
118 }
119 </style>
120 </head>
```

Penjelasan code :

- **Bagian PHP**

- `include("koneksi.php");`: Menggunakan pernyataan `include` untuk menyertakan file `koneksi.php`, yang kemungkinan berisi skrip koneksi ke database MySQL.

- `if(isset($_POST['proses'])) { ... }`: Mengecek apakah formulir telah disubmit dengan mengecek keberadaan parameter POST bernama 'proses'. Ini biasanya diatur oleh tombol submit dengan nama "proses" pada formulir.

Mendapatkan nilai dari formulir menggunakan `$_POST` dan menyimpannya dalam variabel:

- `$id_dokter`: ID Dokter (tipe data number).

- `$nama_dokter`: Nama Dokter (tipe data text).

- `$spesialis`: Spesialis (tipe data text).

- `$alamat`: Alamat (tipe data text).

- `$no_telp`: Nomor Telepon (tipe data number).

- Mengeksekusi query SQL untuk menyisipkan data dokter baru ke dalam tabel "klinik". Hasilnya disimpan dalam variabel `$query`.

- Menggunakan `<script>alert('Data Dokter telah tersimpan')</script>` untuk menampilkan pesan peringatan setelah data dokter disimpan.

- **Bagian HTML**

- `<h3> Tambah dokter </h3>`: Menampilkan judul "Tambah dokter" menggunakan elemen HTML `<h3>`.

- `<form action="" method="POST">...</form>`: Membuat formulir HTML dengan metode POST dan action kosong (mengarah ke halaman yang sama). Data formulir akan dikirimkan ke halaman itu sendiri.

- Tabel HTML untuk menampilkan input fields dan tombol submit. Setiap `<tr>` (baris) mewakili satu atribut dokter (ID dokter, Nama dokter, dll.), dan setiap `<td>` (sel) mewakili input field.

- Pengaturan CSS digunakan untuk memberikan tata letak dan tampilan yang estetik pada halaman HTML dan formulir. Misalnya, memberikan warna latar belakang, margin, padding, dan menentukan lebar formulir.

4. dokter_uibah.php

Penjelasan code :

```
1 <?php
2 include "koneksi.php";
3 $sql=mysql_query($koneksi,"select * from klinik where id_dokter=$GET[idokter]");
4 $data=mysql_fetch_array($sql);
5
6 ?>
7
8 <h3> Ubah Data Dokter </h3>
9
10 <form action="" method="post">
11 <table>
12 <tr>
13 <td width="100%"> ID dokter </td>
14 <td> <input type="text" name="id_dokter" value="<?php echo $data[id_dokter]; ?>" </td>
15 </tr>
16 <tr>
17 <td> Nama dokter </td>
18 <td> <input type="text" name="nama_dokter" value="<?php echo $data[nama_dokter]; ?>" </td>
19 </tr>
20 <tr>
21 <td> Spesialis </td>
22 <td> <input type="text" name="spesialis" value="<?php echo $data[spesialis]; ?>" </td>
23 </tr>
24 <tr>
25 <td> Alamat </td>
26 <td> <input type="text" name="alamat" value="<?php echo $data[alamat]; ?>" </td>
27 </tr>
28 <tr>
29 <td> Nomor telepon </td>
30 <td> <input type="text" name="no_telp" value="<?php echo $data[no_telp]; ?>" </td>
31 </tr>
32 <tr>
33 <td></td>
34 <td><input type="submit" name="proses" value="Ubah"> </td>
35 </tr>
36 </table>
37 </form>
38
39 <?php
40 include "koneksi.php";
41
42
43 if(isset($_POST['proses'])){
44 mysql_query($koneksi,"update klinik set
45 nama_dokter = $_POST[nama_dokter],
46 spesialis = $_POST[spesialis],
47 alamat = $_POST[alamat],
48 no_telp = $_POST[no_telp]
49 where id_dokter = '$GET[idokter]'");
50
51 echo "Data dokter telah diubah";
52 echo "<a href='\"#\"'>refresh content</a>";
53
54 }
55
56 ?>
57
58 <DOCTYPE html>
59 <html lang="en">
60 <head>
61 <meta charset="UTF-8">
62 <meta name="viewport" content="width=device-width, initial-scale=1">
63 <title>Ubah Data Dokter</title>
64
65 <style>
66 body {
67 font-family: Courier, sans-serif;
68 background-image: url(obat.jpg);
69 background-size: cover;
70 margin: 0;
71 padding: 0;
72 display: flex;
73 flex-direction: column;
74 align-items: center;
75 height: 100vh;
76 }
77
78 h3 {
79 color: #333;
80 text-align: center;
81 margin-top: 100px;
82 }
83
84 form {
85 width: 50%;
86 margin: 20px auto;
87 background-color: #fff;
88 padding: 20px;
89 border-radius: 8px;
90 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
91 }
92
93 table {
94 width: 100%;
95 margin-top: 10px;
96 }
97
98 table tr, table td {
99 padding: 10px;
100 border-bottom: 1px solid #ddd;
101 }
102
103 table tr:last-child, form input[type="submit"] {
104 border-bottom: none;
105 }
106
107 input[type="text"] {
108 width: calc(100% - 22px);
109 padding: 5px;
110 margin-top: 5px;
111 margin-bottom: 12px;
112 box-sizing: border-box;
113 }
114
115 input[type="submit"] {
116 width: 100%;
117 padding: 10px;
118 background-color: #4CAF50;
119 color: white;
120 cursor: pointer;
121 border: none;
122 border-radius: 4px;
123 }
124
125 input[type="submit"]:hover {
126 background-color: #4CAF50;
127 }
128 </style>
129 </head>
130 </html>
```

Penjelasan code :

- **Bagian PHP**

Mengambil data untuk diubah :

- include "koneksi.php";: Menggunakan pernyataan include untuk menyertakan file koneksi.php, yang kemungkinan berisi skrip koneksi ke database MySQL.
- Menggunakan parameter \$_GET['kode'] untuk mendapatkan ID dokter yang akan diubah.
- Menjalankan query SQL untuk mengambil data dokter berdasarkan ID dokter yang diambil dari parameter URL.
- Menggunakan mysqli_fetch_array untuk mengambil data sebagai array.

Formulir untuk mengubah data :

- <form action="" method="post"> : Membuka formulir HTML dengan metode POST dan action kosong (mengarah ke halaman yang sama).
- <input type="text" name="id_dokter" value="<?php echo \$data['id_dokter']; ?>"> : Menampilkan nilai awal dari masing-masing input field sesuai dengan data yang diambil dari database.
- if(isset(\$_POST['proses'])) { : Mengecek apakah formulir telah disubmit dengan mengecek keberadaan parameter POST bernama 'proses'. Ini biasanya diatur oleh tombol submit pada formulir.
- mysqli_query(\$koneksi, "update klinik set
nama_dokter = '\$_POST[nama_dokter]',
spesialis = '\$_POST[spesialis]',
alamat = '\$_POST[alamat]',
no_telp = '\$_POST[no_telp]'
where id_dokter = '\$_GET[kode]'"); :

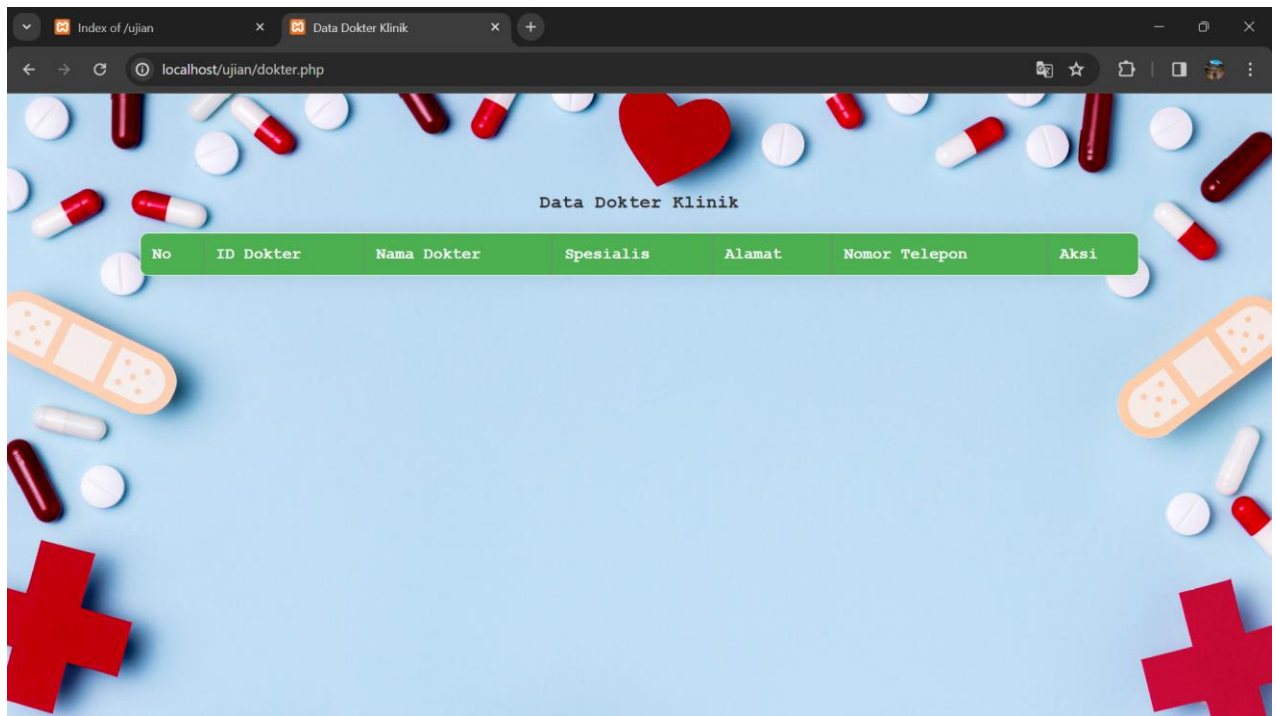
Menjalankan query SQL untuk mengubah data dokter berdasarkan ID dokter yang diambil dari parameter URL.

- echo "Data Dokter telah diubah";
- echo "<meta http-equiv=refresh content=1;URL='dokter.php'>"; :

Menampilkan pesan bahwa data dokter telah diubah dan melakukan pengalihan (refresh) ke halaman "dokter.php" setelah 1 detik.

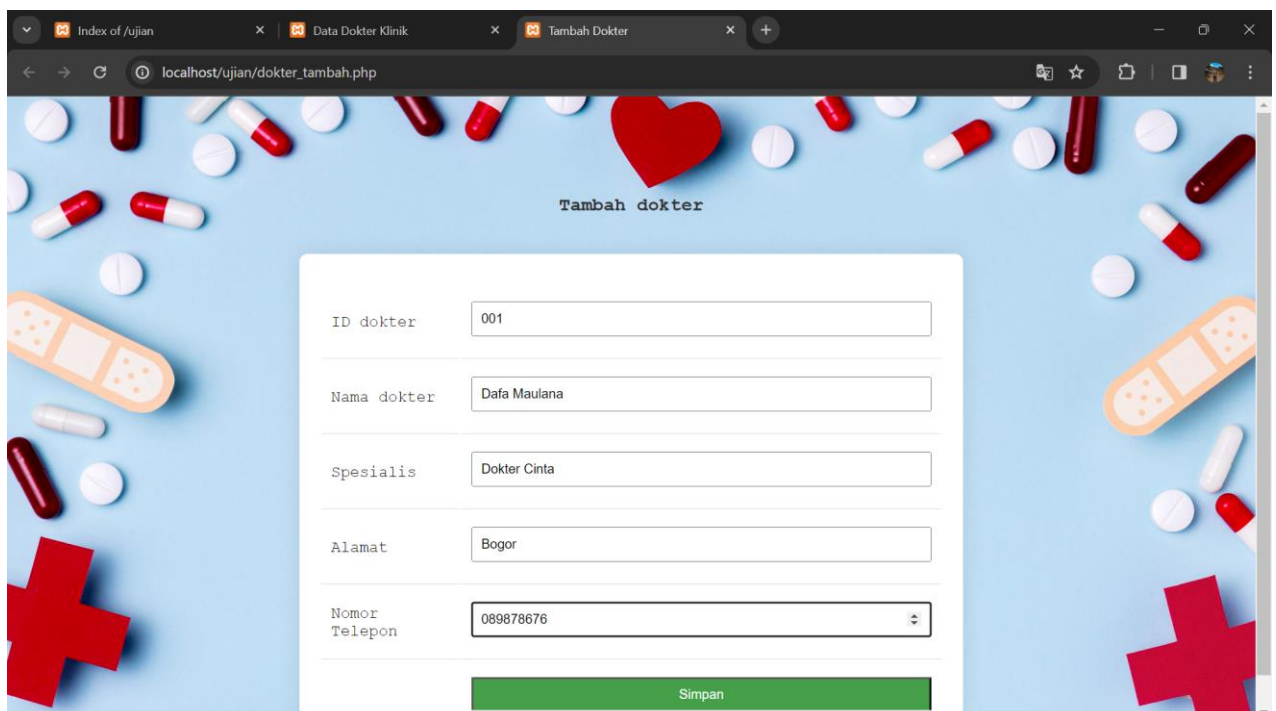
Output Program

1. Tampilan Read Awal



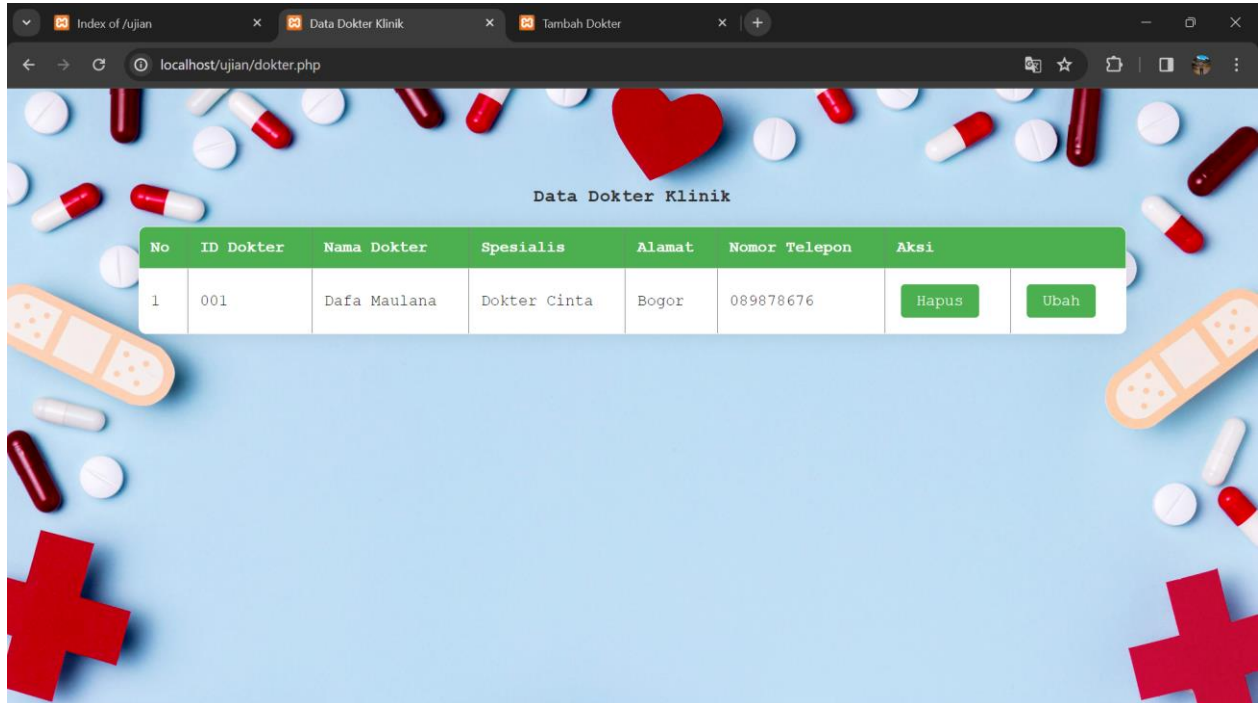
2. Tampilan create

Output create untuk membuat data baru dimana user menginput data pada form dengan menginput ID Dokter, Nama Dokter, Spesialis, Alamat dan Nomor Telepon.



3. Output Read

Laman untuk melihat data yang sudah kita isi pada form create, maka data akan tampil pada halaman utama read.



The screenshot shows a web browser window with the URL `localhost/ujian/dokter.php`. The page title is "Data Dokter Klinik". It displays a table with the following data:

No	ID Dokter	Nama Dokter	Spesialis	Alamat	Nomor Telepon	Aksi
1	001	Dafa Maulana	Dokter Cinta	Bogor	089878676	Hapus Ubah

4. Output Update

Mengubah data pada output yang telah tampil pada halaman utama dengan klik button ubah, yang nantinya akan dikembalikan kembali ke dalam form untuk mengganti data yang ingin diganti. Kemudian klik button ubah untuk menampilkan hasil perubahannya pada halaman utama. Jika bagian ID Dokter yang diupdate maka data tidak akan bisa diupdate, dikarenakan saya memberikan ID Dokter sebagai primary key.

Ubah Data Dokter

ID dokter: 001

Nama dokter: Dafa Maulana

Spesialis: Dokter Hati

Alamat: Bogor

Nomor Telepon: 089878676

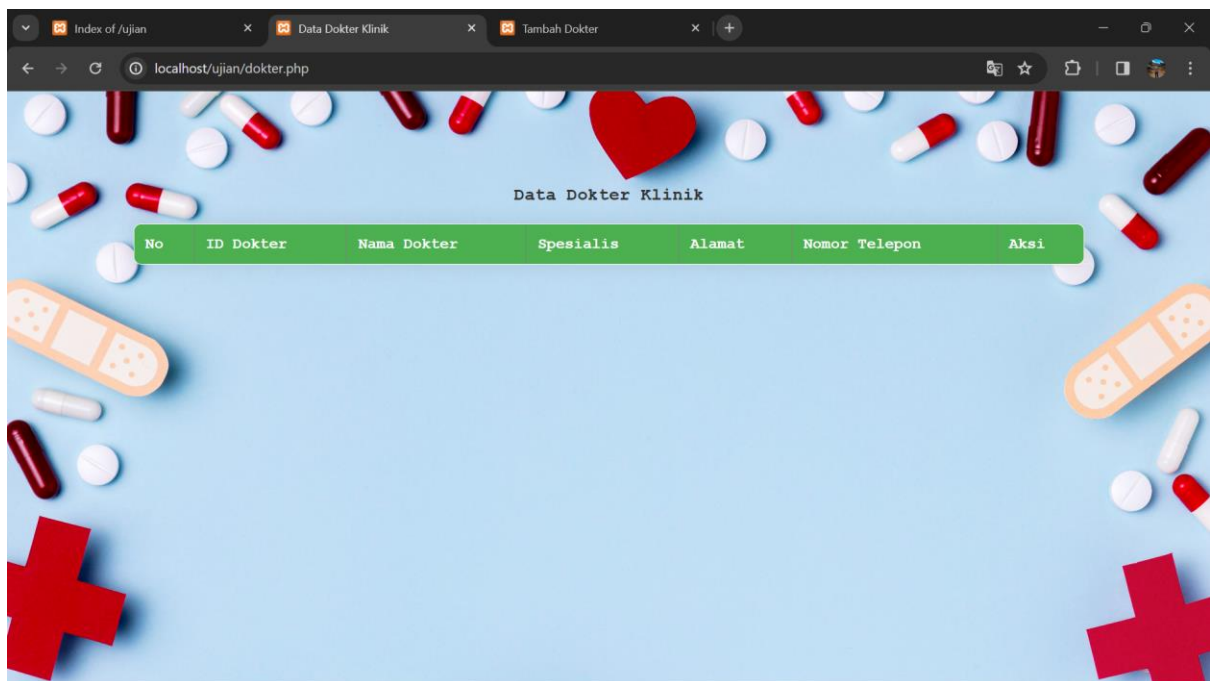
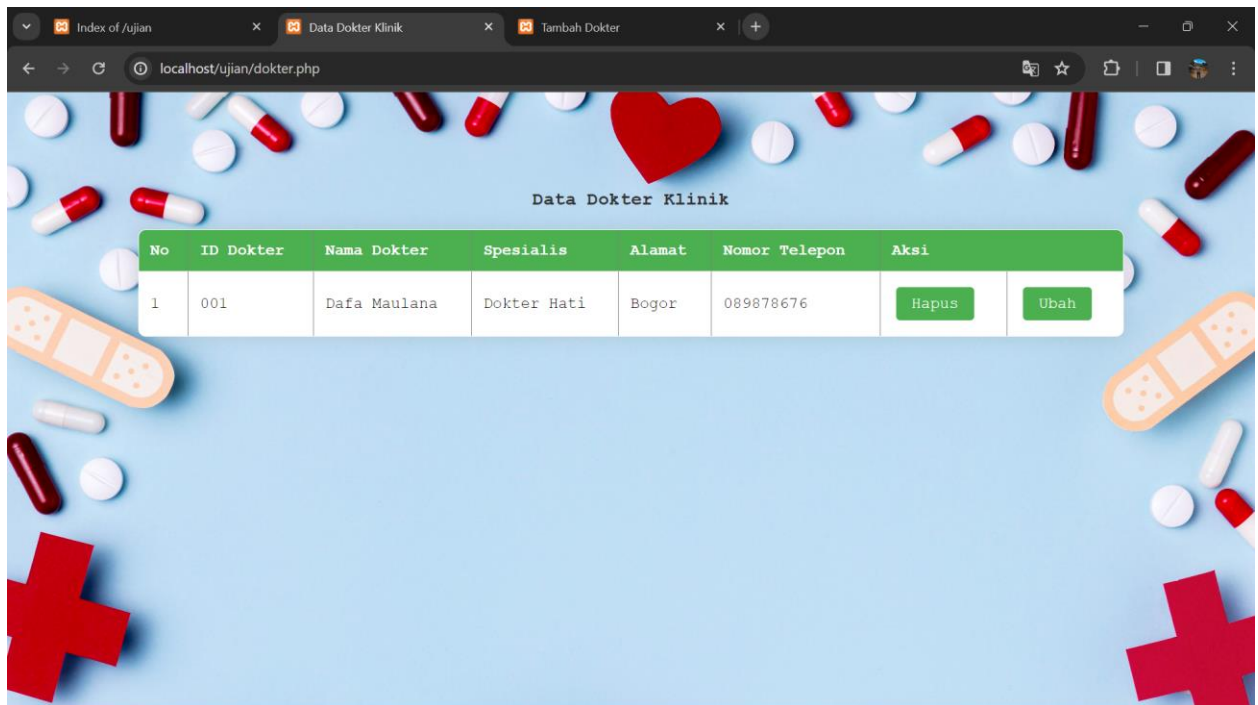
Ubah

Data Dokter Klinik

No	ID Dokter	Nama Dokter	Spesialis	Alamat	Nomor Telepon	Aksi
1	001	Dafa Maulana	Dokter Hati	Bogor	089878676	Hapus Ubah

5. Output Delete

Melakukan proses penghapusan output pada halaman utama dengan menekan button hapus.



6. Output phpMyAdmin setelah input data

The screenshot shows the phpMyAdmin web interface in a browser. The URL is `localhost/phpmyadmin/index.php?route=/sql&db=ujian&table=klinik&pos=0`. The interface displays the 'klinik' table from the 'ujian' database. The table has 5 columns: `id_dokter`, `nama_dokter`, `spesialis`, `alamat`, and `no_telp`. A single row of data is visible, corresponding to 'Dafa Maulana', 'Dokter Cinta', 'Bogor', and '08951234'. The SQL query `SELECT * FROM `klinik`` is shown at the top, and the results are displayed below it. The interface includes a sidebar with a database structure tree, a top navigation bar with various tools, and a bottom console area.

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM `klinik`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	id_dokter	nama_dokter	spesialis	alamat	no_telp
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	001	Dafa Maulana	Dokter Cinta	Bogor	08951234

Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Console

BAB IV

Penutup

4.1 Kesimpulan

Perancangan CRUD data klinik dokter menggunakan HTML, PHP, dan CSS berbasis website memungkinkan pengelolaan data klinik secara efisien. Melalui HTML, struktur halaman web dapat dibangun dengan elemen-elemen yang dibutuhkan. CSS memberikan tampilan yang menarik dan responsif bagi pengguna. Sementara itu, PHP berperan dalam interaksi dengan database, memungkinkan pembuatan, pembacaan, pembaruan, dan penghapusan data klinik (Create, Read, Update, Delete). Dengan integrasi ketiga bahasa pemrograman ini, pengguna dapat dengan mudah menambahkan, melihat, mengubah, atau menghapus data klinik melalui antarmuka web yang user-friendly.