

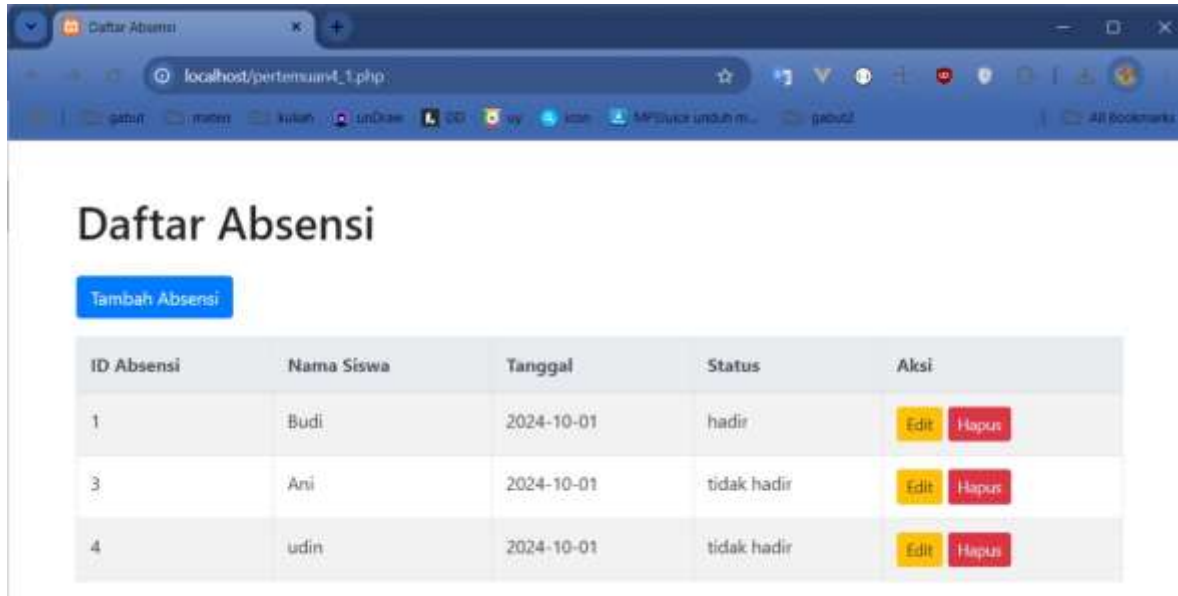
Tugas Dokumentasi CRUD(1)

Kelompok uhuy :

- 225410081_Alfian Desna Saputra
 - 225410089_Arnoldus Paulus Sewa
 - 225410095_Muhammad Khoirul Anam Ihsan
-

Capture gambar output dari Dokumentasi CRUD(1)

Tampilan utama



Tampilan crud input



Input Absensi

Nama Siswa:

Tanggal:

Status:

[Tambah Absensi](#)

Tambah Siswa Baru

Nama Siswa:

[Tambah Siswa](#)

Tampilan edit



Edit Absensi

Status:

[Update](#)

[Kembali](#)

Tampilan output apabila dipencet tombol hapus



Pembahasan:

Pada tugas kali ini mahasiswa diminta untuk Tugas Dokumentasi CRUD(1)

Tampilan utama :

Pertama, program ini terbagi menjadi 2 bagian:

Koneksi ke Database:

Program ini diawali dengan konfigurasi koneksi ke database sekolah.

Informasi yang dibutuhkan seperti host, username, password, dan nama database (sekolah) disimpan dalam variable.

Script kemudian membuat koneksi ke database menggunakan fungsi `mysqli_connect`.

Jika koneksi gagal, maka program akan menampilkan pesan error dan berhenti berjalan.

Tampilan Daftar Absensi:

Program dilanjutkan dengan query untuk membaca data absensi.

Query ini menggunakan inner join untuk menggabungkan tabel absensi dengan tabel siswa berdasarkan `id_siswa`.

Hasil query kemudian disimpan di variable `$result`.

Selanjutnya program akan mengecek apakah ada data absensi yang ditemukan (`$result > num_rows > 0`).

Jika ada data absensi, maka program akan menampilkan tabel berisi daftar absensi.

Tabel ini dibuat dengan menggunakan tag HTML `<table>` dan dibagi menjadi header dan body.

Header tabel berisi judul kolom seperti ID Absensi, Nama Siswa, Tanggal, Status, dan Aksi.

Body tabel berisi data absensi yang diambil dari hasil query. Setiap baris data (`$row`) akan ditampilkan pada baris baru (`<tr>`) di dalam tabel.

Kolom Aksi berisi tombol edit dan hapus untuk setiap data absensi. Tombol edit akan mengarah ke halaman `edit.php` dengan membawa parameter `id_absensi` untuk proses edit data. Tombol hapus akan memunculkan konfirmasi sebelum data dihapus.

Jika tidak ada data absensi, maka program akan menampilkan pesan "0 hasil".

Selanjutnya program menangani aksi hapus data absensi:

Program akan mengecek apakah ada parameter delete pada URL (`isset($_GET['delete'])`).

Jika parameter delete ada, maka program akan mengambil nilai `id_absensi` yang akan dihapus.

Script kemudian membuat query untuk menghapus data absensi berdasarkan `id_absensi`.

Jika query delete berhasil, maka program akan menampilkan pesan sukses dan melakukan refresh halaman (header("Location: pertemuan4_1.php")). Script juga akan dihentikan (exit()) agar tidak ada output tambahan.

Jika query delete gagal, maka program akan menampilkan pesan error beserta detail error dari database.

Terakhir, program akan menutup koneksi ke database.

Secara keseluruhan, program ini merupakan contoh sederhana CRUD (Create, Read, Update, Delete) menggunakan PHP dan database.

Berikut beberapa hal yang bisa dikembangkan dari program ini:

Menambahkan formulir untuk menambah data absensi baru (halaman create.php belum ditampilkan pada kode yang diberikan).

Menambahkan fitur edit data absensi (halaman edit.php belum ditampilkan pada kode yang diberikan).

Menambahkan fitur filter data absensi berdasarkan tanggal, kelas, atau siswa tertentu.

Menampilkan detail informasi siswa pada halaman daftar absensi.

Tampilan crud input

Tambah Absensi Baru:

Program diawali dengan pengecekan method request (\$_SERVER['REQUEST_METHOD']) dan keberadaan parameter create_absensi.

Jika method request adalah POST dan parameter create_absensi ada, maka program akan mengambil nilai dari formulir yaitu id_siswa, tanggal, dan status.

Script kemudian membuat query untuk insert data absensi baru ke dalam tabel absensi.

Jika query insert berhasil (\$conn->query(\$sql) === TRUE), maka program akan menampilkan pesan sukses dan melakukan refresh ke halaman utama (pertemuan4_1.php). Script juga akan dihentikan (exit()).

Jika query insert gagal, maka program akan menampilkan pesan error beserta detail error dari database.

Tambah Siswa Baru:

Program dilanjutkan dengan pengecekan serupa seperti sebelumnya, namun dengan parameter create_siswa.

Jika method request adalah POST dan parameter create_siswa ada, maka program akan mengambil nilai dari formulir yaitu nama_siswa.

Script kemudian membuat query untuk insert data siswa baru ke dalam tabel siswa.

Perlakuan program selanjutnya sama seperti pada saat tambah absensi, yaitu menampilkan pesan sukses atau pesan error tergantung hasil query.

Menampilkan Data Siswa:

Program kemudian melakukan query untuk mengambil semua data siswa dari tabel siswa.

Hasil query disimpan di variable \$result_siswa.

Formulir Tambah Absensi:

Program menampilkan formulir untuk menambah absensi baru.

Formulir ini menggunakan method POST dan action diarahkan ke halaman ini sendiri (action="").

Formulir terdiri dari beberapa input:

Dropdown untuk memilih nama siswa. Dropdown ini diisi dengan data siswa yang diambil dari query sebelumnya.

Input type date untuk memilih tanggal absensi.

Dropdown untuk memilih status absensi (Hadir/Tidak Hadir).

Tombol submit untuk memproses data absensi baru.

Formulir Tambah Siswa:

Program dilanjutkan dengan menampilkan formulir untuk menambah siswa baru.

Formulir ini juga menggunakan method POST dan action diarahkan ke halaman ini sendiri (action="").

Formulir hanya terdiri dari satu input type text untuk nama siswa dan tombol submit untuk memproses data siswa baru.

Penutup:

Program diakhiri dengan menutup koneksi ke database.

Tampilan edit

edit data absensi di database sekolah menggunakan MySQLi. Pertama, kode menghubungkan ke database dan memeriksa apakah koneksi berhasil. Jika pengguna memilih untuk mengedit data, sistem mengambil data absensi berdasarkan id_absensi yang dikirim melalui URL dan menampilkan formulir edit menggunakan Bootstrap.

Tampilan output apabila dipencet tombol hapus

Tombol "Hapus" pada gambar berfungsi untuk menghapus data absensi yang terkait dengan baris tertentu. Saat tombol ditekan, muncul dialog konfirmasi yang bertanya, "Anda yakin ingin menghapus?" untuk memastikan pengguna tidak menghapus data secara tidak sengaja. Pengguna dapat memilih "OK" untuk melanjutkan penghapusan atau "Cancel" untuk membatalkannya. Warna merah pada tombol ini menunjukkan bahwa tindakan tersebut bersifat permanen dan perlu perhatian lebih dari pengguna sebelum melakukannya.