Web toDolist menggunakan php, disini terdapat 2 role user yaitu user dan superuser. Superuser bisa menghapus dan membuat data user baru, sementara user bisa melakukan crud terkait dengan kegiatan mereka.

**Info login:**

**User:**

username: bayu

Password: uas

**Superuser:**

username: admin

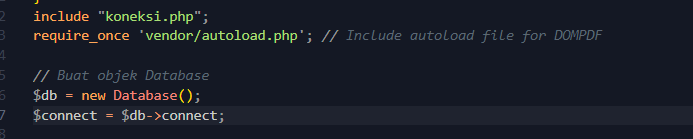
Password: admin

**Konfigurasi database:**

Database menggunakan mysql dengan nama database todolist tanpa password

****

Penggunaan koneksi di file dengan menggunakan objek



**Fungsi-fungsi:**

Fungsi pendaftaran akun baru yang akan mengambil data dari form daftar dan mengecek isi inputan:

<?php

include 'koneksi.php'; *// Sesuaikan dengan file koneksi Anda*

*// Buat objek Database*

$db = new Database();

$connect = $db->connect;

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $username = $\_POST['username'];

    $email = $\_POST['email'];

    $password = md5($\_POST['password']);

    $confirm\_password = md5($\_POST['confirm\_password']);

*// Periksa apakah password dan konfirmasi password cocok*

    if ($password != $confirm\_password) {

        echo "<script>alert('Konfirmasi Password tidak sesuai')</script>";

        echo "<meta http-equiv='refresh' content='1;url=daftar.php'>";

        exit(); *// Menghentikan eksekusi skrip jika konfirmasi password tidak sesuai*

    }

*// Cek apakah username atau email sudah ada di database*

    $checkQuery = "SELECT \* FROM user WHERE username='$username' OR email='$email'";

    $checkResult = mysqli\_query($connect, $checkQuery);

    if (mysqli\_num\_rows($checkResult) > 0) {

        echo "<script>alert('Username atau Email sudah digunakan')</script>";

        echo "<meta http-equiv='refresh' content='1;url=daftar.php'>";

        exit(); *// Menghentikan eksekusi skrip jika username atau email sudah digunakan*

    }

*// Enkripsi password sebelum disimpan ke database*

    $hashed\_password = md5($password);

*// Simpan data ke database*

    $insertQuery = "INSERT INTO user (username, email, password) VALUES ('$username', '$email', '$hashed\_password')";

    $insertResult = mysqli\_query($connect, $insertQuery);

    if ($insertResult) {

        echo "<script>alert('Pendaftaran berhasil')</script>";

        echo "<meta http-equiv='refresh' content='1;url=formLogin.php'>";

        exit(); *// Menghentikan eksekusi skrip setelah pendaftaran berhasil*

    } else {

        echo "<script>alert('Gagal melakukan pendaftaran')</script>";

        echo "<meta http-equiv='refresh' content='1;url=daftar.php'>";

        exit(); *// Menghentikan eksekusi skrip jika gagal melakukan pendaftaran*

    }

} else {

    echo "Akses tidak diizinkan";

}

?>

Fungsi login. kode melakukan pemeriksaan apakah halaman ini diakses melalui metode POST dengan menggunakan $\_SERVER["REQUEST\_METHOD"] == "POST". Jika iya, maka kode akan mengambil nilai dari form login yang dikirim melalui $\_POST, membersihkannya menggunakan mysqli\_real\_escape\_string untuk mencegah SQL injection, dan mengenkripsi password menggunakan fungsi md5.

Selanjutnya, kode menjalankan query SQL untuk mencari pengguna dengan username dan password yang sesuai di database. Jika ditemukan, variabel $check akan berisi jumlah baris hasil query. Jika $check lebih dari 0, artinya pengguna ditemukan dan sesi login akan dimulai.

Dalam sesi login, kita menyimpan beberapa informasi penting seperti user\_id, login status, dan role pengguna. Jika pengguna memiliki peran sebagai superuser, maka akan diarahkan ke halaman dashboard khusus superuser menggunakan header("location: dashboard.php");. Jika bukan superuser, maka akan diarahkan ke halaman listAktivitas.php.

Jika username dan password tidak cocok, maka akan ditampilkan pesan kesalahan menggunakan echo "<script>alert('Username atau Password salah')</script>"; dan pengguna akan diarahkan kembali ke halaman form login setelah 1 detik menggunakan <meta http-equiv='refresh' content='1 url=formLogin.php'>.

<?php

session\_start();

*// Import kelas Database*

include 'koneksi.php';

*// Buat objek Database*

$db = new Database();

$connect = $db->connect;

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $username = mysqli\_real\_escape\_string($connect, $\_POST['username']);

    $password = mysqli\_real\_escape\_string($connect, md5($\_POST['password']));

    $query = "SELECT \* FROM user WHERE username= '$username' AND password = '$password'";

    $queryDB = mysqli\_query($connect, $query);

    $check = mysqli\_num\_rows($queryDB);

    if ($check > 0) {

        $dataUser = mysqli\_fetch\_array($queryDB);

        $\_SESSION['user\_id'] = $dataUser['user\_id']; *// Ambil user\_id dari hasil query*

        $\_SESSION['login'] = true;

        $\_SESSION['role'] = $dataUser['role']; *// Misalnya ada kolom 'role' untuk menentukan superuser atau bukan*

        if ($\_SESSION['role'] == 'superuser') {

            header("location: dashboard.php"); *// Arahkan ke dashboard superuser*

            exit();

        } else {

            header("location: listAktivitas.php"); *// Arahkan ke dashboard user biasa*

            exit();

        }

    } else {

        echo "<script>alert('Username atau Password salah')</script>";

        echo "<meta http-equiv='refresh' content='1 url=formLogin.php'>";

    }

}

?>

Fungsi simpan aktivitas, Kode dimulai dengan memeriksa apakah sesi login telah diatur; jika tidak, pengguna akan diarahkan kembali ke halaman login. Setelah memastikan adanya sesi dan pengiriman data melalui metode POST, kode mengambil nilai input dari form, memvalidasi agar tidak ada data yang kosong, dan menghubungkan ke database melalui kelas Database yang diimpor dari file koneksi.php. Data aktivitas kemudian disimpan ke dalam database dengan menggunakan query INSERT INTO. Jika penyimpanan data berhasil, pengguna diarahkan ke halaman listAktivitas.php; jika tidak, pesan "Gagal Menyimpan Data!!" ditampilkan. Kode juga menangani situasi ketika metode pengiriman data tidak valid dengan menampilkan pesan yang sesuai.

<?php

session\_start();

*// Cek apakah ada sesi login*

if (!isset($\_SESSION['login']) || empty($\_SESSION['login'])) {

    header("location: formLogin.php");

    exit; *// Menghentikan eksekusi skrip jika tidak ada sesi login*

}

*// Memastikan ada input yang dikirimkan*

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

*// Mengambil data aktivitas dari form*

    $aktivitas = $\_POST['aktivitas'];

    $start = $\_POST['start'];

    $end = $\_POST['end'];

    $status = $\_POST['status'];

*// Validasi data agar tidak kosong*

    if (!empty($aktivitas) && !empty($start) && !empty($end) && !empty($status)) {

*// Menghubungkan ke database*

        include "koneksi.php"; *// Pastikan file koneksi.php sesuai dengan konfigurasi database Anda*

        $db = new Database();

        $connect = $db->connect;

*// Mendapatkan user\_id dari sesi*

        if (isset($\_SESSION['user\_id']) && !empty($\_SESSION['user\_id'])) {

            $user\_id = $\_SESSION['user\_id']; *// Pastikan sesuai dengan nama sesi yang digunakan saat login*

*// Menyiapkan query untuk menyimpan aktivitas*

            $sql = "INSERT INTO kegiatan (aktivitas, start, end, status, user\_id)

                    VALUES ('$aktivitas', '$start', '$end', '$status', '$user\_id')";

*// Eksekusi query*

            if (mysqli\_query($connect, $sql)) {

                header('Location: listAktivitas.php');

                exit; *// Menghentikan eksekusi skrip setelah melakukan redirect*

            } else {

                echo "Gagal Menyimpan Data!!";

            }

        } else {

            echo "ID Pengguna Tidak Valid!";

        }

    } else {

        echo "Data tidak boleh kosong!";

    }

} else {

    echo "Metode pengiriman data tidak valid!";

}

?>

Fungsi update Kode tersebut menghubungkan ke database melalui kelas Database yang diimpor dari file koneksi.php, dan menerima nilai dari form update-barang melalui metode POST. Selanjutnya, kode menjalankan perintah SQL untuk mengupdate data kegiatan berdasarkan id\_kegiatan yang diterima dari form. Jika proses update berhasil, pengguna akan diarahkan ke halaman listAktivitas.php; jika tidak, akan ditampilkan pesan "Gagal Update data".

<?php

*//Include file koneksi ke database*

include "koneksi.php";

$db = new Database();

$connect = $db->connect;

*//menerima nilai dari kiriman form update-barang*

$id\_kegiatan=$\_POST["id\_kegiatan"];

$aktivitas=$\_POST["aktivitas"];

$start=$\_POST["start"];

$end=$\_POST["end"];

$status=$\_POST["status"];

*//perintah sql untuk update data barang berdasarkan id\_barang yang akan di update*

    $sql="update kegiatan set

        aktivitas='$aktivitas',

        start='$start',

        end='$end',

        status='$status'

        where id\_kegiatan=$id\_kegiatan";

*//Eksekusi query sql diatas*

    $hasil=mysqli\_query($connect,$sql);

*//kondisi apabila gagal maka akan tampil pemberitahuan gagal*

        if (!$hasil) {

            echo "Gagal Update data";

            exit;

        }

*//apabila berhasil maka halaman akan di redirect ke tampil-barang.php*

    header("Location:listAktivitas.php");

?>

Fungsi hapus digunakan untuk menghapus data kegiatan dari database berdasarkan id\_kegiatan yang diterima dari URL. Setelah menghubungkan ke database melalui kelas Database yang diimpor dari file koneksi.php, kode menangkap nilai id\_kegiatan yang dikirim melalui URL menggunakan $\_GET['id\_kegiatan']. Selanjutnya, data kegiatan dengan id tersebut dihapus dari database dengan menggunakan perintah SQL DELETE. Setelah penghapusan data berhasil dilakukan, pengguna akan diarahkan kembali ke halaman listAktivitas.php menggunakan fungsi header("location:listAktivitas.php").

<?php

*// koneksi database*

include 'koneksi.php';

$db = new Database();

$connect = $db->connect;

*// menangkap data id yang di kirim dari url*

$id = $\_GET['id\_kegiatan'];

*// menghapus data dari database*

mysqli\_query($connect,"delete from kegiatan where id\_kegiatan='$id'");

*// mengalihkan halaman kembali ke index.php*

header("location:listAktivitas.php");

?>

**Library dan package yang digunakan:**

Bootstrap

Fontawesome

dompdf