Nama: Muhammad Naufal Dzakwan
NIM: H1D023094
Kelas: Pemrograman Web II

Tugas 1-Pemrograman Web

1. Penjelasan Kode
    a. db.php
    kode ini berfungsi untuk Menghubungkan ke Database Menggunakan PDO untuk membuat koneksi dengan parameter host, nama database, username, dan password serta menangani Error Menggunakan try-catch untuk menangkap dan menampilkan error jika koneksi gagal.
    Potongan kodenya:

```
<?php
class Database {
   private $host = "localhost";
   private $db_name = "blog_db";
   private $username = "root";
   private $password = "";
   public $conn;

   public function getConnection() {
      $this->conn = null;
      try {
         $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" .
$this->db_name, $this->username, $this->password);
         $this->conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
      } catch (PDOException $exception) {
         echo "Connection error: " . $exception->getMessage();
      }
      return $this->conn;
   }
}
?>
```

    b. register.php
    Kode ini berfungsi untuk Memproses Registrasi User dengan menerima input username, email, dan password melalui form, Memanggil Kelas User seperti data yang dikirim akan diproses menggunakan metode register() dari kelas User untuk menyimpan user baru ke database, serta Menampilkan Notifikasi saat registrasi berhasil, lalu user diarahkan ke halaman login, sedangkan jika gagal, akan muncul pesan error.
    Potongan kodenya:

```
<?php
```

```
session_start();
require_once 'User.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $user = new User();
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    if ($user->register($username, $email, $password)) {
        echo "<script>alert('Registrasi berhasil! Silakan login.');
window.location.href='login.php';</script>";
    } else {
        echo "<script>alert('Registrasi gagal, coba lagi.');</script>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Register</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
    <h2>Register</h2>
    <form method="POST">
        <div class="mb-3">
            <label>Username:</label>
            <input type="text" name="username" required class="form-control">
        </div>
        <div class="mb-3">
            <label>Email:</label>
            <input type="email" name="email" required class="form-control">
        </div>
        <div class="mb-3">
            <label>Password:</label>
            <input type="password" name="password" required class="form-
control">
        </div>
```

```
        <button type="submit" class="btn btn-primary">Register</button>
        <a href="login.php">Sudah punya akun? Login</a>
    </form>
</body>
</html>
```

c. login.php

Kode ini menangani autentikasi user dengan menerima input email dan password dari form login. Data yang dikirim akan diperiksa menggunakan metode login() dari kelas User, lalu mencocokkannya dengan database. Jika login berhasil, sesi user dibuat dan diarahkan ke index.php, sedangkan jika gagal, muncul pesan error.

Potongan kodenya:

```
<?php
session_start();
require_once 'User.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $user = new User();
    $email = $_POST['email'];
    $password = $_POST['password'];

    $loggedInUser = $user->login($email, $password);
    if ($loggedInUser) {
        $_SESSION['user_id'] = $loggedInUser['id'];
        $_SESSION['username'] = $loggedInUser['username'];
        header('Location: index.php');
        exit();
    } else {
        echo "<script>alert('Login gagal. Periksa email atau password
Anda.');</script>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
<body class="container mt-5">
  <h2>Login</h2>
  <form method="POST">
    <div class="mb-3">
      <label>Email:</label>
      <input type="email" name="email" required class="form-control">
    </div>
    <div class="mb-3">
      <label>Password:</label>
      <input type="password" name="password" required class="form-
control">
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
    <a href="register.php">Belum punya akun? Register</a>
  </form>
</body>
</html>
```

d. user.php

Kode ini mendefinisikan kelas User yang terhubung ke database melalui kelas Database, mendaftarkan user baru dengan Metode register() menyimpan user ke database dengan password yang di-hash untuk keamanan, serta melakukan verifikasi login dengan Metode login() memeriksa email dan mencocokkan password yang diinput dengan data yang tersimpan di database menggunakan password_verify()

Potongan kodenya:

```
<?php
require_once 'db.php';

class User {
  private $conn;

  public function __construct() {
    $database = new Database();
    $this->conn = $database->getConnection();
  }

  public function register($username, $email, $password) {
    $hashedPassword = password_hash($password,
PASSWORD_DEFAULT);
    $query = "INSERT INTO users (username, email, password) VALUES
(:username, :email, :password)";
```

```
        $stmt = $this->conn->prepare($query);
        return $stmt->execute([':username' => $username, ':email' => $email,
':password' => $hashedPassword]);
    }

    public function login($email, $password) {
        $query = "SELECT * FROM users WHERE email = :email";
        $stmt = $this->conn->prepare($query);
        $stmt->execute([':email' => $email]);
        $user = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($user && password_verify($password, $user['password'])) {
            return $user;
        }
        return false;
    }
}
?>
```

e. logout.php
Kode ini menangani proses logout dengan menghentikan sesi pengguna menggunakan session_destroy(), lalu mengarahkan mereka kembali ke halaman login.
Potongan kodenya:

```
<?php
session_start();
session_destroy();
header("Location: login.php");
exit();
?>
```

f. create_post.php
Kode ini memastikan hanya pengguna yang sudah login (dengan user_id di sesi) yang dapat mengakses halaman pembuatan artikel, Menyimpan Artikel Baru yaitu Data judul dan konten yang diinput oleh pengguna dikirim ke metode createPost() dalam kelas Post untuk disimpan ke database, serta Menampilkan Notifikasi saat artikel berhasil dibuat, lalu pengguna diarahkan ke halaman utama, sedangkan jika gagal, muncul pesan error.
Potongan kodenya:

```
<?php
session_start();
require_once 'Post.php';

if (!isset($_SESSION['user_id'])) {
```

```
    header('Location: login.php');
    exit();
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $post = new Post();
    $title = $_POST['title'];
    $content = $_POST['content'];
    $user_id = $_SESSION['user_id'];

    if ($post->createPost($title, $content, $user_id)) {
        echo "<script>alert('Artikel berhasil dibuat!');
window.location.href='index.php';</script>";
    } else {
        echo "<script>alert('Gagal membuat artikel.');</script>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Buat Artikel</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
    <h2>Buat Artikel Baru</h2>
    <form method="POST">
        <div class="mb-3">
            <label>Judul:</label>
            <input type="text" name="title" required class="form-control">
        </div>
        <div class="mb-3">
            <label>Konten:</label>
            <textarea name="content" required class="form-control"
rows="5"></textarea>
        </div>
        <button type="submit" class="btn btn-primary">Buat Artikel</button>
    </form>
</body>
```

```
</html>
```

g. post.php

kode ini mengelola Koneksi Database contohnya seperti Kelas Post terhubung ke database melalui kelas Database untuk mengelola data artikel. Kode ini menyediakan metode untuk membuat (createPost()), membaca (getPosts(), getPost()), memperbarui (updatePost()), dan menghapus (deletePost()) artikel dalam database dan kode ini Menghubungkan Artikel dengan Pengguna dengan Metode getPosts() mengambil data artikel beserta nama pengguna yang membuatnya dengan menggunakan JOIN pada tabel users.

Potongan kodenya:

```
<?php
require_once 'db.php';

class Post {
    private $conn;

    public function __construct() {
        $database = new Database();
        $this->conn = $database->getConnection();
    }

    public function createPost($title, $content, $user_id) {
        $query = "INSERT INTO posts (title, content, user_id) VALUES (:title,
:content, :user_id)";
        $stmt = $this->conn->prepare($query);
        return $stmt->execute([':title' => $title, ':content' => $content, ':user_id'
=> $user_id]);
    }

    public function getPosts() {
        $query = "SELECT posts.*, users.username FROM posts JOIN users ON
posts.user_id = users.id ORDER BY created_at DESC";
        return $this->conn->query($query)->fetchAll(PDO::FETCH_ASSOC);
    }

    public function getPost($id) {
        $query = "SELECT * FROM posts WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        $stmt->execute([':id' => $id]);
        return $stmt->fetch(PDO::FETCH_ASSOC);
    }
```

```
    public function updatePost($id, $title, $content) {
        $query = "UPDATE posts SET title = :title, content = :content WHERE
id = :id";
        $stmt = $this->conn->prepare($query);
        return $stmt->execute([':title' => $title, ':content' => $content, ':id' =>
$id]);
    }

    public function deletePost($id) {
        $query = "DELETE FROM posts WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        return $stmt->execute([':id' => $id]);
    }
}
?>
```

h. comment.php

Kode ini mendefinisikan kelas Comment untuk mengelola komentar pada sebuah postingan dengan menghubungkan ke database melalui kelas Database. Metode addComment() digunakan untuk menambahkan komentar ke dalam database berdasarkan post_id dan user_id. Sedangkan metode getComments() mengambil daftar komentar dari database yang terkait dengan sebuah postingan, termasuk informasi username dari tabel users, lalu mengurutkannya berdasarkan waktu pembuatan.

potongan kodenya:

```
<?php
require_once 'db.php';

class Comment {
    private $conn;

    public function __construct() {
        $database = new Database();
        $this->conn = $database->getConnection();
    }

    public function addComment($post_id, $user_id, $content) {
        $query = "INSERT INTO comments (post_id, user_id, content) VALUES
(:post_id, :user_id, :content)";
        $stmt = $this->conn->prepare($query);
        return $stmt->execute([':post_id' => $post_id, ':user_id' => $user_id,
':content' => $content]);
    }
```

```
    public function getComments($post_id) {
        $query = "SELECT comments.*, users.username FROM comments JOIN
    users ON comments.user_id = users.id WHERE post_id = :post_id ORDER
    BY created_at DESC";
        $stmt = $this->conn->prepare($query);
        $stmt->execute([':post_id' => $post_id]);
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
?>
```

i. view.php

Kode ini menampilkan halaman detail sebuah artikel beserta komentar yang terkait dengan artikel tersebut. Data artikel diambil menggunakan metode getPost() dari kelas Post, sedangkan komentar diambil dengan metode getComments() dari kelas Comment. Jika pengguna yang sudah login mengirim komentar melalui form, komentar akan disimpan ke database menggunakan addComment(), lalu halaman akan dimuat ulang untuk menampilkan komentar terbaru.

potongan kodenya:

```
<?php
session_start();
require_once 'Post.php';
require_once 'Comment.php';

if (!isset($_GET['id'])) {
    header('Location: index.php');
    exit();
}

$post = new Post();
$comment = new Comment();
$article = $post->getPost($_GET['id']);
$comments = $comment->getComments($_GET['id']);

if (!$article) {
    echo "<script>alert('Artikel tidak ditemukan.');
window.location.href='index.php';</script>";
}

if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_SESSION['user_id'])) {
```

```
    $comment->addComment($_GET['id'], $_SESSION['user_id'],
$_POST['content']);
    header("Location: view.php?id=" . $_GET['id']);
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title><?= htmlspecialchars($article['title']) ?></title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
    <h2><?= htmlspecialchars($article['title']) ?></h2>
    <p><small>Diposting pada: <?= $article['created_at'] ?></small></p>
    <p><?= nl2br(htmlspecialchars($article['content'])) ?></p>

    <hr>
    <h4>Komentar</h4>
    <?php if (isset($_SESSION['user_id'])): ?>
        <form method="POST">
            <div class="mb-3">
                <textarea name="content" required class="form-control"
rows="3"></textarea>
            </div>
            <button type="submit" class="btn btn-primary">Tambah
Komentar</button>
        </form>
    <?php else: ?>
        <p><a href="login.php">Login</a> untuk berkomentar.</p>
    <?php endif; ?>

    <ul class="list-group mt-3">
        <?php foreach ($comments as $c): ?>
            <li class="list-group-item">
                <strong><?= htmlspecialchars($c['username']) ?>:</strong>
                <p><?= nl2br(htmlspecialchars($c['content'])) ?></p>
                <small><?= $c['created_at'] ?></small>
            </li>
```

```
        <?php endforeach; ?>
    </ul>
</body>
</html>
```

j. index.php

Kode ini menampilkan daftar artikel yang tersimpan dalam database dengan mengambil data melalui metode getPosts() dari kelas Post. Jika pengguna sudah login, tombol untuk membuat artikel (create_post.php) akan ditampilkan. Data artikel ditampilkan dalam tabel menggunakan DataTables agar lebih interaktif dan mudah digunakan. Setiap artikel memiliki tombol "View" yang mengarahkan pengguna ke halaman detail artikel (view.php). Selain itu, kode ini juga memuat pustaka Bootstrap untuk tampilan yang lebih menarik dan jQuery untuk mengaktifkan fitur DataTables.

potongan kodenya:

```
<?php
session_start();
require_once 'Post.php';
$post = new Post();
$posts = $post->getPosts();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Blog</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
    <h2 class="mb-3">Blog Posts</h2>
        <?php if (isset($_SESSION['user_id'])): ?>
    <a href="create_post.php" class="btn btn-success mb-3">Buat Artikel</a>
<?php endif; ?>

    <table id="postsTable" class="table table-striped">
        <thead>
            <tr>
                <th>Title</th>
```

```
                <th>Author</th>
                <th>Date</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($posts as $p): ?>
                <tr>
                    <td><?= htmlspecialchars($p['title']) ?></td>
                    <td><?= htmlspecialchars($p['username']) ?></td>
                    <td><?= $p['created_at'] ?></td>
                    <td><a href="view.php?id=<?= $p['id'] ?>" class="btn btn-
primary btn-sm">View</a></td>
                </tr>
            <?php endforeach; ?>
        </tbody>
    </table>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script
src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>
    <script>
        $(document).ready(function() {
            $('#postsTable').DataTable();
        });
    </script>
</body>
</html>
```

2. Link github
   https://github.com/DzakwanGoreng/TugasPemwebII