

# **LAPORAN AKHIR TUGAS BESAR**

## **PEMROGRAMAN WEB**



**Disusun Oleh :**

- Marcellinus Yovian Indrastata	1202210019
- Rafi Ardinata Riskiansyah	1202212028
- Axel Danu Pramudita	1202210011
- Maoreen Damar Safira	1202210001
- Khofifah Dharmasari	1202210045

**Dosen Pengampu:**

**Riza Akhsani Setyo Prayoga, S.Kom,  
M.MT**

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNOLOGI INFORMASI DAN**  
**BISNIS INSTITUT TEKNOLOGI TELKOM**  
**SURABAYA 2023**

## LEMBAR KONTRIBUSI Pengerjaan Tugas Kelompok

**DOSEN : Riza Akhsani Setyo Prayoga, S.Kom, M.MT**

No	NIM	Nama Mahasiswa	Kontribusi
1	1202210019	Marcellinus Yovian Indrastata	<ul style="list-style-type: none"><li>- Membuat source code create, view, dan login</li><li>- Membuat desain dari isi website admin (create, add, delete,login)</li><li>- Memberi Saran dan Kesimpulan Pada Praktikum yang Di Kerjakan</li></ul>
2	1202212028	Rafi Ardinata Riskiansyah	<ul style="list-style-type: none"><li>- Membuat database dan menggabungkan semua source code</li><li>- Membuat desain dari isi website admin (create, add, delete,login)</li><li>- Memberi Saran dan Kesimpulan Pada Praktikum yang Di Kerjakan serta Menyusun laporan praktikum.</li></ul>
3	1202210011	Axel Danu Pramudita	<ul style="list-style-type: none"><li>- Membuat desain dari isi website admin (create, add, delete,login)</li><li>- Membuat source code create, view, dan login</li><li>- Memberi Saran dan Kesimpulan Pada Praktikum yang Di Kerjakan serta Menyusun laporan praktikum.</li></ul>
4	1202210001	Maoreen Damar Safira	<ul style="list-style-type: none"><li>- Membuat desain dari isi website admin (create, add, delete,login)</li><li>- Membuat source code create, view, dan login</li><li>- Memberi Saran dan Kesimpulan Pada Praktikum yang Di Kerjakan serta Menyusun laporan praktikum.</li></ul>
5	1202210045	Khofifah Dharmasari	<ul style="list-style-type: none"><li>- Membuat desain dari isi website admin (create, add, delete,login)</li><li>- Membuat source code create, view, dan login</li><li>- Memberi Saran dan Kesimpulan Pada Praktikum yang Di Kerjakan serta Menyusun laporan praktikum.</li></ul>

## **A. PENDAHULUAN**

Pendidikan merupakan salah satu sektor penting dalam masyarakat, karena memiliki peran besar dalam membentuk generasi penerus bangsa yang cerdas dan berkualitas. Dalam era digital dan teknologi informasi seperti saat ini, pendidikan juga mengalami transformasi yang signifikan. Salah satu aspek penting dalam pendidikan adalah pemanfaatan teknologi untuk meningkatkan efisiensi dan efektivitas proses belajar mengajar.

Pada tugas besar akhir UAS ini, kelompok kami mendapatkan kesempatan untuk melanjutkan proyek yang telah dimulai sebelumnya, yaitu pembuatan sebuah situs web tentang pendidikan yang bertema Lexa Institut, sebuah universitas yang berfokus pada memberikan layanan pendidikan berkualitas. Sebelumnya, pada tugas UTS, fokus utama kelompok kami adalah pada bagian front-end, yaitu merancang tampilan antarmuka (UI) situs web dan mengisinya dengan materi menggunakan CMS (Content Management System) dan HTML.

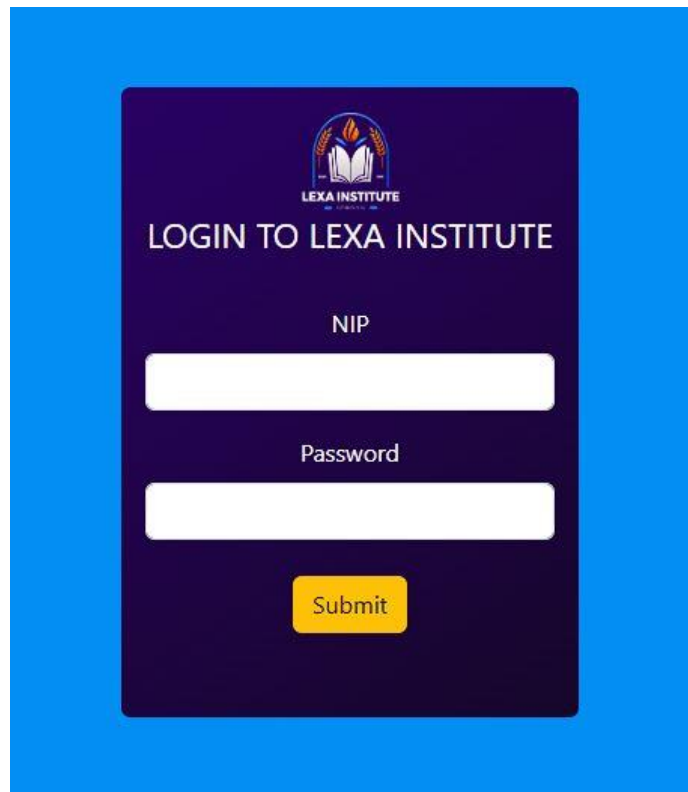
Untuk tugas besar akhir UAS ini, kelompok kami disuruh untuk melanjutkan pengembangan situs web tersebut dengan menambahkan sisi admin yang bertujuan untuk mengatur bagian back-end. Hal ini akan melibatkan penerapan teknologi seperti PHP, Bootstrap, dan fungsionalitas CRUD (Create, Retrieve, Update, Delete) pada database yang terintegrasi dengan situs web. Dengan adanya fitur CRUD, para admin akan memiliki kemampuan untuk membuat, mengambil, memperbarui, dan menghapus data yang terkait dengan konten situs web, seperti artikel, pengumuman, kurikulum, dan lain sebagainya.

Pengembangan sisi admin dengan fitur CRUD ini menjadi krusial karena memberikan kelengkapan pada situs web Lexa Institut dan memungkinkan para admin untuk dengan mudah mengelola konten yang ada. Selain itu, penerapan teknologi PHP dan Bootstrap akan meningkatkan fleksibilitas, kemudahan, dan keamanan dalam mengelola back-end situs web tersebut.

Dengan demikian, tugas besar akhir ini akan menjadi kesempatan bagi kelompok kami untuk mengasah kemampuan dalam pengembangan web secara menyeluruh, meliputi tampilan antarmuka yang menarik (front-end) dan juga bagian back-end yang berperan penting dalam mengelola konten dan data situs web. Semoga melalui tugas besar akhir ini, situs web Lexa Institut dapat menjadi sebuah platform pendidikan yang bermanfaat dan efisien bagi para pengguna dan adminnya.

## B. PEMBAHASAN

- LOGIN



```
1 <?php
2
3 $title = "Lexa Institute | Login";
4
5 // dd($database);
6
7 if (isset($_POST['submit'])) {
8     $nip = htmlentities(strip_tags(trim($_POST["nip"])));
9     $password = htmlentities(strip_tags(trim($_POST["password"])));
10
11     $error_message = "";
12     if (empty($nip) || empty($password)) {
13         $error_message .= "NIP or Password is required";
14     }
15
16     // $nip = mysqli_real_escape_string($connection, $nip);
17     // $password = mysqli_real_escape_string($connection, $password);
18
19     $password_sha1 = sha1($password);
20
21     $user = $database->query("select * from users where nip = ? and password = ?", [$nip, $password_sha1])->fetch();
22
23     if ($user) {
24         $_SESSION['nip'] = $nip;
25         header("Location: /admin/dashboard");
26         exit();
27     } else {
28         $error_message = "NIP or Password is incorrect";
29     }
30 } else {
31     $email = "";
32     $password = "";
33     $error_message = "";
34 }
35
36 // $error_message = [];
37
38 require "resources/views/Admin/Auth/Login.view.php";
39
```

Source code di atas untuk mengelola proses login untuk halaman admin sebuah website atau aplikasi. Mari kita bahas bagian-bagian dari kode ini:

1. ``$title``: Variabel ini digunakan untuk menyimpan judul halaman, yaitu "Lexa Institute | Login". Judul ini kemungkinan akan digunakan untuk ditampilkan pada halaman HTML atau sebagai informasi tambahan.
2. Proses Login:
  - Pada bagian berikutnya, kode melakukan pengecekan apakah data login telah dikirimkan melalui metode POST dengan memeriksa apakah ada data yang dikirimkan dengan tombol bernama "submit". Jika ada, maka proses login akan dijalankan.
  - Variabel ``$nip`` dan ``$password`` digunakan untuk menyimpan data NIP (Nomor Induk Pegawai) dan password yang diambil dari inputan form. Data ini kemungkinan akan digunakan untuk proses autentikasi.
3. Validasi Data:

Setelah mengambil data NIP dan password dari form, kode melakukan validasi untuk memastikan bahwa kedua field tersebut tidak kosong. Jika salah satu atau keduanya kosong, maka variabel ``$error_message`` akan berisi pesan "NIP or Password is required".
4. Enkripsi Password:

Password yang diterima dari form tidak disimpan dalam bentuk plaintext. Kode menggunakan fungsi ``sha1()`` untuk mengenkripsi password sebelum mencocokkannya dengan password yang telah dienkripsi di dalam database.
5. Query ke Database:
  - Kode melakukan query ke database dengan menggunakan variabel ``$database`` (mungkin adalah objek koneksi ke database). Query ini mencari data pengguna (users) berdasarkan NIP dan password yang telah dienkripsi sebelumnya.
  - Jika data pengguna ditemukan (artinya, query mengembalikan hasil), maka proses login berhasil. Variabel sesi ``$_SESSION['nip']`` diset untuk menyimpan NIP pengguna yang berhasil login, dan pengguna akan diarahkan ke halaman "/admin/dashboard".
6. Penanganan Kesalahan Login:

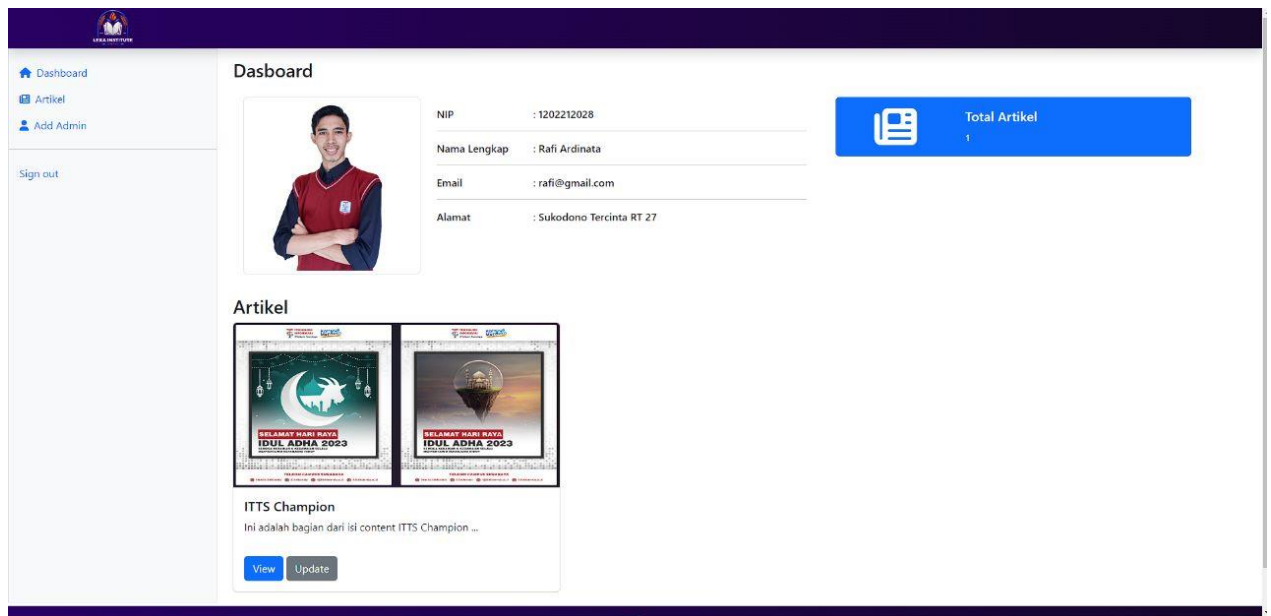
Jika proses login gagal karena NIP atau password tidak cocok dengan data yang ada di database, maka variabel ``$error_message`` akan diisi dengan pesan "NIP or Password is incorrect".
7. Pengaturan Variabel untuk Form:

Jika belum ada data yang dikirimkan melalui form, maka variabel ``$email``, ``$password``, dan ``$error_message`` akan diatur dengan nilai default (kosong).
8. Pemanggilan View:

Terakhir, kode akan memuat tampilan halaman login yang berada di file "resources/views/Admin/Auth/Login.view.php". Dengan menggunakan view ini, tampilan login akan ditampilkan ke pengguna.

Fungsi keseluruhan dari kode ini adalah untuk mengelola proses login pengguna ke halaman admin, dengan mengambil data NIP dan password dari form, melakukan validasi, mengenkripsi password, dan memverifikasi data tersebut di dalam database sebelum mengizinkan pengguna masuk ke halaman dashboard jika proses login berhasil. Jika terjadi kesalahan, pesan kesalahan akan ditampilkan pada halaman login.

- **Dashboard**



```
1 <?php
2
3 if (!isset($_SESSION['nip'])) {
4     header("Location: /admin/login");
5     exit();
6 }
7
8 // dd($_SESSION['nip']);
9
10 $title = "LEXA INSTITUTE | DASBOARD";
11 $style = "/resources/css/dashboard.css";
12
13 $nip = $_SESSION['nip'];
14
15 $users = $database->getData("SELECT * FROM `users` WHERE `nip` = $nip ;");
16 $user = $users[0];
17
18 $articles = $database->getData("SELECT * FROM `posts` WHERE `nip` = $nip ;");
19 $numArticles = count($articles);
20 // dd($numArticles);
21
22
23 // dd($user);
24
25
26 require("resources/views/Admin/home/home.view.php");
27
```

Sourve code ini untuk menampilkan halaman dashboard (dasbor) pada halaman admin sebuah website atau aplikasi berbasis PHP. Mari kita bahas bagian-bagian dari kode ini:

1. Pemeriksaan Sesi (Session) dan Pengalihan Halaman

- Kode ini pertama-tama memeriksa apakah variabel sesi `\$\_SESSION['nip']` sudah di-set atau ada. `\$\_SESSION['nip']` kemungkinan digunakan untuk menyimpan data NIP (Nomor Induk Pegawai) pengguna yang sedang login.
- Jika variabel sesi `\$\_SESSION['nip']` tidak di-set (artinya pengguna belum login), maka pengguna akan diarahkan kembali ke halaman login admin ("/admin/login") menggunakan fungsi `header()`. Kemudian, perintah `exit()` digunakan untuk menghentikan eksekusi kode selanjutnya dan memastikan pengalihan halaman terjadi

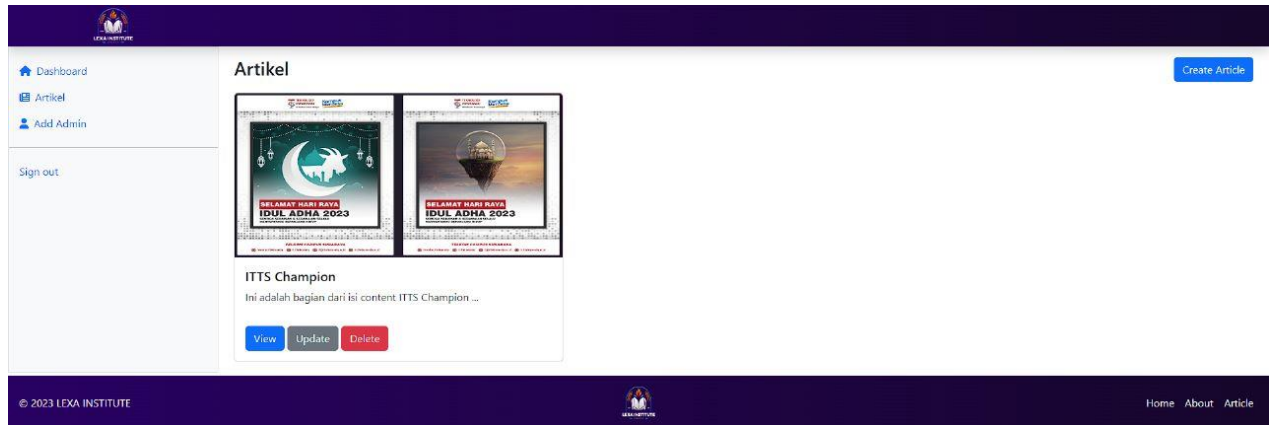
2. Penyiapan Variabel untuk Tampilan:

- Selanjutnya, kode ini menyiapkan beberapa variabel untuk tampilan halaman dashboard ("resources/views/Admin/home/home.view.php")

- Variabel `$title` berisi judul halaman yang akan ditampilkan pada tampilan.
  - Variabel `$style` berisi path untuk file CSS yang akan digunakan pada halaman tersebut.
3. Mendapatkan Data Pengguna (User) dan Artikel (Posts) Pengguna:
- Kode ini mengambil data pengguna dan data artikel yang telah dibuat oleh pengguna (user) berdasarkan NIP yang disimpan dalam variabel `$_SESSION['nip']`.
  - Data pengguna diambil dari tabel "users" menggunakan perintah `SELECT` dan disimpan dalam variabel `$user`. Kode ini mengasumsikan bahwa hanya ada satu baris data pengguna yang sesuai dengan NIP tersebut, dan data tersebut akan disimpan dalam bentuk array.
  - Data artikel diambil dari tabel "posts" menggunakan perintah `SELECT` dan disimpan dalam variabel `$articles`. Kode ini mengasumsikan bahwa pengguna dapat membuat beberapa artikel, sehingga data artikel akan disimpan dalam bentuk array.
  - Jumlah artikel yang telah dibuat oleh pengguna dihitung menggunakan fungsi `count()` dan disimpan dalam variabel `$numArticles`.
4. Pemanggilan View:
- Terakhir, kode ini akan memuat tampilan halaman dashboard ("`resources/views/Admin/home/home.view.php`"). Data-data yang telah diambil sebelumnya, seperti data pengguna dan data artikel, akan digunakan dalam tampilan ini untuk menampilkan informasi yang sesuai.

Fungsi keseluruhan dari kode ini adalah untuk menampilkan halaman dashboard yang berisi informasi tentang pengguna yang sedang login (seperti nama, email, dan alamat) serta daftar artikel yang telah dibuat oleh pengguna. Jumlah total artikel yang telah dibuat juga ditampilkan di halaman ini. Halaman dashboard ini memberikan informasi penting bagi pengguna sebagai ringkasan aktivitas dan informasi pribadi mereka di halaman admin.

- **CRUD**



```

1  <?php
2
3  if (!isset($_SESSION['nip'])) {
4      header("Location: /admin/login");
5      exit();
6  }
7
8  // dd($_SESSION['nip']);
9
10 $title = "LEXA INSTITUTE | ARTICLES";
11 $style = "/resources/css/dashboard.css";
12
13 $nip = $_SESSION['nip'];
14 $users = $database->getData("SELECT * FROM `users` WHERE `nip` = $nip ;");
15 $user = $users[0];
16
17
18 $articles = $database->getData("SELECT * FROM `posts` WHERE `nip` = $nip ;");
19
20 // dd($result);
21
22 require("resources/views/Admin/Article/index.view.php");
23

```

Source code ini menampilkan halaman daftar artikel (posts) yang telah dibuat oleh seorang pengguna (user) pada halaman admin sebuah website atau aplikasi berbasis PHP. Mari kita bahas bagian-bagian dari kode ini:

1. Pemeriksaan Sesi (Session) dan Pengalihan Halaman:

- Kode ini pertama-tama memeriksa apakah variabel sesi `\$\_SESSION['nip']` sudah di-set atau ada. `\$\_SESSION['nip']` kemungkinan digunakan untuk menyimpan data NIP (Nomor Induk Pegawai) pengguna yang sedang login.
- Jika variabel sesi `\$\_SESSION['nip']` tidak di-set (artinya pengguna belum login), maka pengguna akan diarahkan kembali ke halaman login admin ("/admin/login") menggunakan fungsi `header()`. Kemudian, perintah `exit()` digunakan untuk menghentikan eksekusi kode selanjutnya dan memastikan pengalihan halaman terjadi.

2. Penyiapan Variabel untuk Tampilan:

- Selanjutnya, kode ini menyiapkan beberapa variabel untuk tampilan halaman daftar artikel ("resources/views/Admin/Article/index.view.php")
- Variabel `\$title` berisi judul halaman yang akan ditampilkan pada tampilan.
- Variabel `\$style` berisi path untuk file CSS yang akan digunakan pada halaman tersebut.

3. Mendapatkan Data Pengguna (User):

- Kode ini mengambil data pengguna dari tabel "users" berdasarkan NIP yang disimpan dalam variabel `\$\_SESSION['nip']`.
- Data pengguna diambil dari database menggunakan perintah `SELECT` dan disimpan



dalam variabel `\$user`. Kode ini mengasumsikan bahwa hanya ada satu baris data pengguna yang sesuai dengan NIP tersebut, dan data tersebut akan disimpan dalam bentuk array.

4. Mendapatkan Data Artikel (Posts) Pengguna:

- Kode ini mengambil data artikel yang telah dibuat oleh pengguna (user) berdasarkan NIP yang disimpan dalam variabel `\$\_SESSION['nip']`.
- Data artikel diambil dari database menggunakan perintah `SELECT` dan disimpan dalam variabel `\$articles`. Kode ini mengasumsikan bahwa pengguna dapat membuat beberapa artikel, sehingga data artikel akan disimpan dalam bentuk array.

5. Pemanggilan View:

Terakhir, kode ini akan memuat tampilan halaman daftar artikel ("resources/views/Admin/Article/index.view.php"). Data-data yang telah diambil sebelumnya, seperti data pengguna dan data artikel, akan digunakan dalam tampilan ini untuk menampilkan informasi yang sesuai.

Fungsi keseluruhan dari kode ini adalah untuk menampilkan halaman daftar artikel yang telah dibuat oleh pengguna (user) yang sedang login. Kode ini akan mengambil data pengguna dari tabel "users", mengambil data artikel dari tabel "posts", dan menampilkan informasi tersebut pada halaman yang sesuai dengan tampilan "index.view.php". Pengguna akan dapat melihat daftar artikel yang telah mereka buat melalui halaman ini.

The screenshot displays a web application interface for creating a user account. The main content area is titled "CREATE USER ACCOUNT" and contains several input fields: "Nomor Identitas Pegawai Negeri Sipil (NIP)", "Name", "Email", "Password", "Alamat", and "Image User Photo". The "Image User Photo" field includes a "Choose File" button and a "No file chosen" status. At the bottom of the form are "Submit" and "Cancel" buttons. On the left, a sidebar menu lists "Dashboard", "Artikel", "Add Admin", and "Sign out". The footer of the page includes the Lexa Institute logo, the text "© 2023 LEXA INSTITUTE", and navigation links for "Home", "About", and "Article".



```
1  <?php
2
3  class User
4  {
5      private $nip;
6      private $name;
7      private $email;
8      private $alamat;
9      private $image_path;
10
11     public function __construct($nip, $name, $email, $alamat, $image_path)
12     {
13         $this->nip = $nip;
14         $this->name = $name;
15         $this->email = $email;
16         $this->alamat = $alamat;
17         $this->image_path = $image_path;
18     }
19     public function getNip()
20     {
21         return $this->nip;
22     }
23
24     public function setNip($nip)
25     {
26         $this->nip = $nip;
27     }
28
29     public function getName()
30     {
31         return $this->name;
32     }
33
34     public function setName($name)
35     {
36         $this->name = $name;
37     }
38
39     public function getEmail()
40     {
41         return $this->email;
42     }
43
44     public function setEmail($email)
45     {
46         $this->email = $email;
47     }
48
49     public function getAlamat()
50     {
51         return $this->alamat;
52     }
53
54     public function setAlamat($alamat)
55     {
56         $this->alamat = $alamat;
57     }
58
59     public function getImage_path()
60     {
61         return $this->image_path;
62     }
63
64     public function setImage_path($image_path)
65     {
66         $this->image_path = $image_path;
67     }
68 }
69
```

Source code user ini berfungsi sebagai representasi atau model untuk menyimpan data seorang pengguna (user) dengan atribut-atributnya seperti NIP (Nomor Induk Pegawai), nama, email, alamat, dan path gambar profil.

1. Atribut Kelas:

- Kelas "User" memiliki beberapa atribut (properties) yang digunakan untuk menyimpan data seorang pengguna. Atribut-atribut tersebut adalah:

- ``$nip``: Menyimpan NIP (Nomor Induk Pegawai) seorang pengguna.
- ``$name``: Menyimpan nama lengkap seorang pengguna.
- ``$email``: Menyimpan alamat email seorang pengguna.
- ``$alamat``: Menyimpan alamat lengkap seorang pengguna
- ``$image_path``: Menyimpan path (lokasi) gambar profil seorang pengguna

2. Konstruktor:

- Kelas "User" memiliki konstruktor yang digunakan untuk membuat objek "User" baru dengan memberikan nilai awal pada atribut-atributnya. Konstruktor ini dipanggil saat membuat objek baru dari kelas "User".
- Konstruktor menerima beberapa parameter yaitu ``$nip``, ``$name``, ``$email``, ``$alamat``, dan ``$image_path``. Nilai-nilai ini digunakan untuk menginisialisasi nilai awal dari atribut-atribut objek

3. Metode Getters dan Setters:

- Kelas "User" memiliki beberapa metode getters dan setters yang digunakan untuk mengakses dan mengubah nilai atribut-atribut objek.
- Metode getters digunakan untuk mengambil nilai atribut dan metode setters digunakan untuk mengubah nilai atribut.

Dengan menggunakan kelas "User" ini, kita dapat dengan mudah membuat objek untuk mewakili seorang pengguna dengan informasi seperti NIP, nama, email, alamat, dan path gambar profilnya. Selain itu, dengan adanya metode getters dan setters, kita dapat dengan mudah mengakses dan mengubah nilai atribut-atribut tersebut tanpa perlu langsung mengakses atributnya secara langsung.

Kelas "User" ini sangat membantu dalam mengatur dan menyimpan data seorang pengguna, dan juga memberikan metode-metode untuk mengakses dan mengubah data tersebut dengan cara yang terstruktur dan terorganisir.



## CREATE ARTICLE

Title

Tagline

Open this select menu



Content

Image

Choose File

No file chosen

Submit

Cancel



```
1 <?php
2
3 if (isset($_SESSION['nip'])) {
4     header("Location: /admin/login");
5     exit();
6 }
7
8 // dd($_SESSION['nip']);
9
10 $title = "LEXA INSTITUTE | CREATE ARTICLES";
11 $style = "/resources/css/dashboard.css";
12
13
14 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
15     // dd($_POST);
16     $errors = [];
17     // dd($_POST);
18     $articleTitle = htmlentities(strip_tags(trim($_POST["article_title"])));
19     $tagline = htmlentities(strip_tags(trim($_POST["tagline"])));
20     $content = htmlentities(strip_tags(trim($_POST["article_content"])));
21
22     // Check if a file was uploaded successfully
23     if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
24         // The temporary location of the uploaded file
25         $tmpFilePath = $_FILES['image']['tmp_name'];
26
27         // Generate a unique name for the file to avoid overwriting
28         $imageFileName = uniqid() . '_' . $_FILES['image']['name'];
29
30         // The destination path to move the uploaded file to the 'article_photos' directory
31         $destinationFilePath = $_SERVER['DOCUMENT_ROOT'] . '/publics/article_photos/' . $imageFileName;
32         // dd($destinationFilePath);
33
34         // Move the uploaded file to the new location
35         if (move_uploaded_file($tmpFilePath, $destinationFilePath)) {
36             // File was successfully moved, save the image path in the database
37             $image = $imageFileName;
38         } else {
39             // There was an error moving the file, handle the error accordingly
40             $errors['image'] = "Failed to upload the image. Please try again.";
41         }
42     } else {
43         // No file was uploaded or an error occurred during upload
44         $errors['image'] = "Please select an image to upload.";
45     }
46
47     if (empty($articleTitle)) {
48         $errors['article_title'] = "Please enter a title of Article";
49     }
50     if (empty($tagline)) {
51         $errors['tagline'] = "Please enter a tagline of Article";
52     }
53     if (empty($content)) {
54         $errors['article_content'] = "Please enter a content of Article";
55     }
56
57     if (empty($errors)) {
58         // If there are no errors, insert data into the database, including the image path
59         $database->query(
60             "INSERT INTO 'posts'('nip', 'title', 'tagline', 'content', 'image_path')
61             VALUES (?, ?, ?, ?, ?)",
62             [$SESSION['nip'], $articleTitle, $tagline, $content, $image]
63         );
64
65         header("Location: /admin/articles");
66     }
67 }
68
69
70 require("resources/views/Admin/Article/create.view.php");
71
```

Source code ini untuk mengelola proses pembuatan artikel (post) baru pada halaman admin sebuah website atau aplikasi berbasis PHP. Mari kita bahas bagian-bagian dari kode ini:

1. Pemeriksaan Sesi (Session) dan Pengalihan Halaman:
  - Kode ini pertama-tama memeriksa apakah variabel sesi ``$_SESSION['nip']`` sudah di-set atau ada. ``$_SESSION['nip']`` kemungkinan digunakan untuk menyimpan data NIP (Nomor Induk Pegawai) pengguna yang sedang login.
  - Jika variabel sesi ``$_SESSION['nip']`` tidak di-set (artinya pengguna belum login), maka pengguna akan diarahkan kembali ke halaman login admin ("`/admin/login`") menggunakan fungsi ``header()``. Kemudian, perintah ``exit()`` digunakan untuk menghentikan eksekusi kode selanjutnya dan memastikan pengalihan halaman terjadi.
2. Penyiapan Variabel untuk Tampilan:
  - Selanjutnya, kode ini menyiapkan beberapa variabel untuk tampilan halaman pembuatan artikel ("`resources/views/Admin/Article/create.view.php`").
  - Variabel ``$title`` berisi judul halaman yang akan ditampilkan pada tampilan.
  - Variabel ``$style`` berisi path untuk file CSS yang akan digunakan pada halaman tersebut.
3. Pemeriksaan Metode Permintaan (Request Method):

Seperti sebelumnya, kode ini memeriksa apakah permintaan (request) yang diterima adalah melalui metode POST. Ini berarti kode ini hanya akan dieksekusi saat ada data yang dikirimkan melalui form menggunakan metode POST.
4. Pemeriksaan dan Validasi Data:
  - Kode ini melakukan pemeriksaan terhadap data yang dikirimkan melalui form untuk pembuatan artikel baru.
  - Variabel ``$articleTitle``, ``$tagline``, dan ``$content`` masing-masing digunakan untuk menyimpan judul artikel, tagline artikel, dan isi artikel. Data ini diambil dari form dan kemudian di-strip dari tag HTML dan karakter-karakter khusus menggunakan fungsi ``htmlentities()``, ``strip_tags()``, dan ``trim()`` untuk mencegah serangan XSS dan membersihkan data dari spasi yang tidak diinginkan.
5. Mengelola File Gambar (Jika Ada):

Kode ini juga mengelola proses unggah file gambar seperti pada kode sebelumnya. Jika ada file gambar yang diunggah, maka gambar akan disimpan dalam direktori "`publics/article_photos/`" dengan nama yang unik.
6. Validasi Data Lanjutan:

Setelah mengelola file gambar (jika ada), kode ini melakukan validasi lanjutan untuk memastikan bahwa judul artikel, tagline, dan isi artikel tidak kosong. Jika ada data yang kosong, pesan kesalahan akan ditambahkan ke dalam array ``$errors`` sesuai dengan masing-masing field.
7. Menyimpan Data ke Database:
  - Jika tidak ada kesalahan (artinya array ``$errors`` kosong), kode ini akan menyimpan data artikel yang baru dibuat ke dalam database. Data yang disimpan termasuk NIP pengguna yang sedang login, judul artikel, tagline, isi artikel, dan path gambar.
  - Perintah ``INSERT INTO`` digunakan untuk menyisipkan data ke dalam tabel "`posts`" dengan nilai-nilai yang sesuai.
8. Pengalihan Halaman: Jika tidak ada kesalahan dan data berhasil disimpan ke database, pengguna akan diarahkan kembali ke halaman daftar artikel ("`/admin/articles`").

Fungsi keseluruhan dari kode ini adalah untuk mengelola proses pembuatan artikel baru yang diinputkan oleh pengguna melalui form pada halaman admin. Kode ini melakukan validasi data, mengelola unggahan gambar, dan menyimpan data baru ke dalam database jika tidak ada kesalahan. Jika terjadi kesalahan, pesan kesalahan akan ditampilkan untuk memberi tahu pengguna apa yang harus diperbaiki. Setelah artikel berhasil dibuat, pengguna akan diarahkan kembali ke halaman daftar artikel.



## UPDATE ARTICLE

Title  
ITTS Champion

Tagline  
Achievements

Content  
Ini adalah bagian dari isi content ITTS Champion

Image  
Choose File No file chosen



```
1 <?php
2
3 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
4     $errors = [];
5     $id = $_POST['id'];
6     // dd($_POST);
7     $articleTitle = htmlentities(strip_tags(trim($_POST["article_title"])));
8     $tagline = htmlentities(strip_tags(trim($_POST["tagline"])));
9     $content = htmlentities(strip_tags(trim($_POST["article_content"])));
10
11     // Check if a file was uploaded successfully
12     if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
13         // The temporary location of the uploaded file
14         $tmpFilePath = $_FILES['image']['tmp_name'];
15
16         // Generate a unique name for the file to avoid overwriting
17         $imageFileName = uniqid() . '_' . $_FILES['image']['name'];
18
19         // The destination path to move the uploaded file to the 'article_photos' directory
20         $destinationFilePath = $_SERVER['DOCUMENT_ROOT'] . '/publics/article_photos/' . $imageFileName;
21         // dd($destinationFilePath);
22
23         // Move the uploaded file to the new location
24         if (move_uploaded_file($tmpFilePath, $destinationFilePath)) {
25             // File was successfully moved, save the image path in the database
26             $image = $imageFileName;
27         } else {
28             // There was an error moving the file, handle the error accordingly
29             $errors['image'] = "Failed to upload the image. Please try again.";
30         }
31     } else {
32         // No file was uploaded or an error occurred during upload
33         $errors['image'] = "Please select an image to upload.";
34     }
35
36     if (empty($articleTitle)) {
37         $errors['article_title'] = "Please enter a title of Article";
38     }
39     if (empty($tagline)) {
40         $errors['tagline'] = "Please enter a tagline of Article";
41     }
42     if (empty($content)) {
43         $errors['article_content'] = "Please enter a content of Article";
44     }
45
46     if (empty($errors)) {
47         // If there are no errors, insert data into the database, including the image path
48         $nip = $_SESSION['nip'];
49         $database->query(
50             "UPDATE `posts` SET
51                 title = ?,
52                 tagline = ?,
53                 content = ?,
54                 image_path = ?
55                 WHERE id = ?"
56             ,[$articleTitle, $tagline, $content, $image, $id]);
57
58         header("Location: /admin/articles");
59     }
60 }
```

Source code ini untuk mengelola proses pengeditan artikel (post) pada halaman admin sebuah website atau aplikasi berbasis PHP. Mari kita bahas bagian-bagian dari kode ini:

1. **Memeriksa Metode Permintaan (Request Method):**  
Pada bagian awal, kode ini memeriksa apakah permintaan (request) yang diterima adalah melalui metode POST. Ini berarti kode ini hanya akan dieksekusi saat ada data yang dikirimkan melalui form menggunakan metode POST.
2. **Pemeriksaan dan Validasi Data:**
  - Kode ini melakukan pemeriksaan terhadap data yang dikirimkan melalui form untuk mengedit artikel.
  - Variabel ``$id`` diambil dari data ``$_POST['id']``. Ini kemungkinan merupakan ID artikel yang akan diupdate.
  - Variabel ``$articleTitle``, ``$tagline``, dan ``$content`` masing-masing digunakan untuk menyimpan judul artikel, tagline artikel, dan isi artikel. Data ini diambil dari form dan kemudian di-strip dari tag HTML dan karakter-karakter khusus menggunakan fungsi ``htmlentities()``, ``strip_tags()``, dan ``trim()`` untuk mencegah serangan XSS dan membersihkan data dari spasi yang tidak diinginkan.
3. **Mengelola File Gambar (Jika Ada):**
  - Kode ini memeriksa apakah ada file gambar yang diunggah melalui form dengan memeriksa ``$_FILES['image']['error']``. Jika tidak ada error (``UPLOAD_ERR_OK``), artinya file berhasil diunggah.
  - Jika ada file gambar yang diunggah, kode ini memberikan nama unik untuk file gambar dengan menggabungkan ``uniqid()`` (untuk menghasilkan ID unik berdasarkan waktu) dan nama file asli. File gambar ini akan disimpan dalam direktori "publics/article\_photos/" di server dengan menggunakan fungsi ``move_uploaded_file()``.
  - Jika ada masalah saat mengunggah file gambar, pesan kesalahan akan ditambahkan ke dalam array ``$errors['image']``.
4. **Validasi Data Lanjutan:**  
Setelah mengelola file gambar (jika ada), kode ini melakukan validasi lanjutan untuk memastikan bahwa judul artikel, tagline, dan isi artikel tidak kosong. Jika ada data yang kosong, pesan kesalahan akan ditambahkan ke dalam array ``$errors`` sesuai dengan masing-masing field.
5. **Menyimpan Data ke Database:**
  - Jika tidak ada kesalahan (artinya array ``$errors`` kosong), kode ini akan menyimpan data yang diubah ke dalam database. Pengupdatean dilakukan pada tabel "posts" berdasarkan ID artikel yang diambil dari ``$id``.
  - Data judul artikel, tagline, isi artikel, dan path gambar akan diupdate sesuai dengan yang diinputkan oleh pengguna.
6. **Pengalihan Halaman:**  
Jika tidak ada kesalahan dan data berhasil disimpan ke database, pengguna akan diarahkan kembali ke halaman ``/admin/articles``.

Fungsi keseluruhan dari kode ini adalah untuk mengelola proses pengeditan artikel (post) yang diinputkan oleh pengguna melalui form. Kode ini melakukan validasi data, mengelola unggahan gambar, dan menyimpan perubahan ke dalam database jika tidak ada kesalahan. Jika terjadi kesalahan, pesan kesalahan akan ditampilkan untuk memberi tahu pengguna apa yang harus diperbaiki.



Ini adalah bagian dari isi content ITTS Champion

## Article Category

[achievements](#) [Research](#) [Student Exchange](#) [Teaching Assistant Program](#)



```
1 <?php require('resources/views/partials/header.php') ?>
2
3 <?php require('resources/views/partials/nav.php') ?>
4 <main>
5 <?php
6 $date = date("d F Y", strtotime($post['updated_at']));
7 ?>
8 <div class="bg-root">
9 <div class="text-center">
10 <h1 class="text-light pt-5 pb-2">
11 <?=> $post['title'] ?>
12 </h1>
13 <p class="text-light pb-5">
14 <?=> $date ?> / <?=> $post['tagline'] ?>
15 </p>
16 </div>
17 </div>
18 <div class="container px-7 py-3 text-center">
19 <div class="mb-2">
20  $post['title'] ?>" class="rounded" style="max-width: 400px;">
21 </div>
22 <div class="mb-2">
23 <p><?=> $post['content'] ?></p>
24 </div>
25 <div class="mb-2">
26 <h1 class="mb-3">Article Category</h1>
27 <div>
28 <a href="#" class="badge rounded-pill text-bg-primary me-1 text-decoration-none px-3 py-2" role="button"> achievements </a>
29 <a href="#" class="badge rounded-pill text-bg-primary me-1 text-decoration-none px-3 py-2" role="button"> Research </a>
30 <a href="#" class="badge rounded-pill text-bg-primary me-1 text-decoration-none px-3 py-2" role="button"> Student Exchange </a>
31 <a href="#" class="badge rounded-pill text-bg-primary me-1 text-decoration-none px-3 py-2" role="button"> Teaching Assistant Program </a>
32 </div>
33 </div>
34 </div>
35 </main>
36
37 <?php require('resources/views/partials/footer.php') ?>
38
```

Sourve code ini berfungsi untuk menampilkan halaman detail sebuah artikel (post) pada website atau aplikasi berbasis PHP. Mari kita bahas bagian-bagian dari kode ini:

### 1. Inklusi File Partial:

- Kode ini menggunakan beberapa perintah `require()` untuk memasukkan file-file partial yang akan digunakan dalam tampilan halaman. Dua file partial yang dimasukkan adalah "header.php" dan "nav.php" pada bagian atas halaman, serta "footer.php" pada bagian bawah halaman.
- File-file partial ini berisi bagian-bagian yang dapat digunakan secara berulang pada berbagai halaman, seperti header, navigasi, dan footer

### 2. Data Artikel:

Kode ini mengakses data artikel yang akan ditampilkan pada halaman ini. Data artikel tersebut mungkin telah diambil dari database sebelumnya dan disimpan dalam variabel `\$post`

### 3. Tampilan Detail Artikel:

- Kode ini menggunakan data artikel yang telah diakses untuk menampilkan informasi detail tentang artikel tersebut pada halaman.
- Bagian pertama menampilkan judul artikel (`\$post['title']`) dan tanggal terakhir diperbarui (`\$date`) serta tagline artikel (`\$post['tagline']`).
- Bagian selanjutnya menampilkan gambar artikel yang telah diunggah (`\$post['image\_path']`) dengan batasan lebar maksimum 400px.
- Bagian isi artikel ditampilkan menggunakan tag `

` dengan konten dari



``$post['content']``.

- Terakhir, halaman menampilkan kategori artikel dengan menggunakan tautan-tautan yang membentuk badge (gelang) dengan berbagai kategori (misalnya "achievements", "Research", dll.). Namun, tautan-tautan ini tidak berfungsi (`href="#"` artinya hanya placeholder) dan kategori artikel belum diambil dari database, sehingga tidak menampilkan kategori yang sebenarnya.

#### 4. File Partial Footer:

Kode ini menggunakan perintah ``require()`` untuk memasukkan file partial "footer.php" yang berisi bagian footer halaman. Footer ini mungkin berisi informasi tambahan atau tautan ke halaman lain.

Fungsi keseluruhan dari kode ini adalah untuk menampilkan halaman detail sebuah artikel. Halaman ini akan menampilkan informasi seperti judul artikel, tanggal terakhir diperbarui, tagline, gambar artikel, isi artikel, dan kategori artikel. Kode ini merupakan bagian dari tampilan (view) halaman detail artikel dan akan mengambil data dari variabel ``$post`` yang kemungkinan telah diambil sebelumnya dari database atau sumber data lainnya.

	id	nip	title	tagline	content	image_path	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1202212028	ITTS Champion	achievements	Ini adalah bagian dari isi content ITTS Champion	64c09af916b8b_Eid Adha.png	2023-07-26 11:03:05	2023-07-26 11:03:05
<input type="checkbox"/> Check all	With selected: <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete <input type="checkbox"/> Export							

```

1 <?php
2
3 class Database
4 {
5     private $dbHost;
6     private $dbUsername;
7     private $dbPassword;
8     private $dbName;
9     private $connection;
10
11     public function __construct($config)
12     {
13         $this->dbHost = $config['host'];
14         $this->dbUsername = $config['username'];
15         $this->dbPassword = $config['password'];
16         $this->dbName = $config['database'];
17         $this->connection = mysqli_connect($this->dbHost, $this->dbUsername, $this->dbPassword);
18
19         if (!$this->connection) {
20             die("Connection failed: " . mysqli_connect_errno() . " - " . mysqli_connect_error());
21         }
22     }
23
24     public function createDB()
25     {
26         $query = "CREATE DATABASE IF NOT EXISTS " . $this->dbName;
27         $result = mysqli_query($this->connection, $query);
28         if (!$result) {
29             die("Query Error: " . mysqli_errno($this->connection) . " - " . mysqli_error($this->connection));
30         }
31     }
32
33     public function selectDB()
34     {
35         $result = mysqli_select_db($this->connection, $this->dbName);
36         if (!$result) {
37             die("Query Error: " . mysqli_errno($this->connection) . " - " . mysqli_error($this->connection));
38         }
39     }
40
41     public function createTable($tableName, $query)
42     {
43         $result = mysqli_query($this->connection, "CREATE TABLE IF NOT EXISTS $tableName ($query)");
44
45         if (!$result) {
46             die("Query failed: " . mysqli_errno($this->connection) . " - " . mysqli_error($this->connection));
47         }
48     }
49
50     public function query($query, $params = [])
51     {
52         $stmt = $this->connection->prepare($query);
53         $stmt->execute($params);
54
55         return $stmt;
56     }
57
58     public function getData($query)
59     {
60         $result = mysqli_query($this->connection, $query);
61         $resultSet = [];
62
63         if (!$result) {
64             die("Query Error: " . mysqli_errno($this->connection) . " - " . mysqli_error($this->connection));
65         }
66
67         while ($row = mysqli_fetch_array($result)) {
68             $resultSet[] = $row;
69         }
70
71         return $resultSet;
72     }
73
74     public function getConnection()
75     {
76         return $this->connection;
77     }
78 }
79
80 $database = new Database($config);
81 $connection = $database->getConnection();
82 $database->createDB();
83 $database->selectDB();
84
85 $database->createTable("users", "
86     `nip` char(10) NOT NULL,
87     `name` varchar(255) NOT NULL,
88     `email` varchar(255) NOT NULL,
89     `password` varchar(255) NOT NULL,
90     `alamat` text,
91     `image_path` varchar(255) NOT NULL,
92     `isAdmin` tinyint(1) NOT NULL,
93     `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
94     `updated_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
95     PRIMARY KEY (`nip`)
96 ");

```

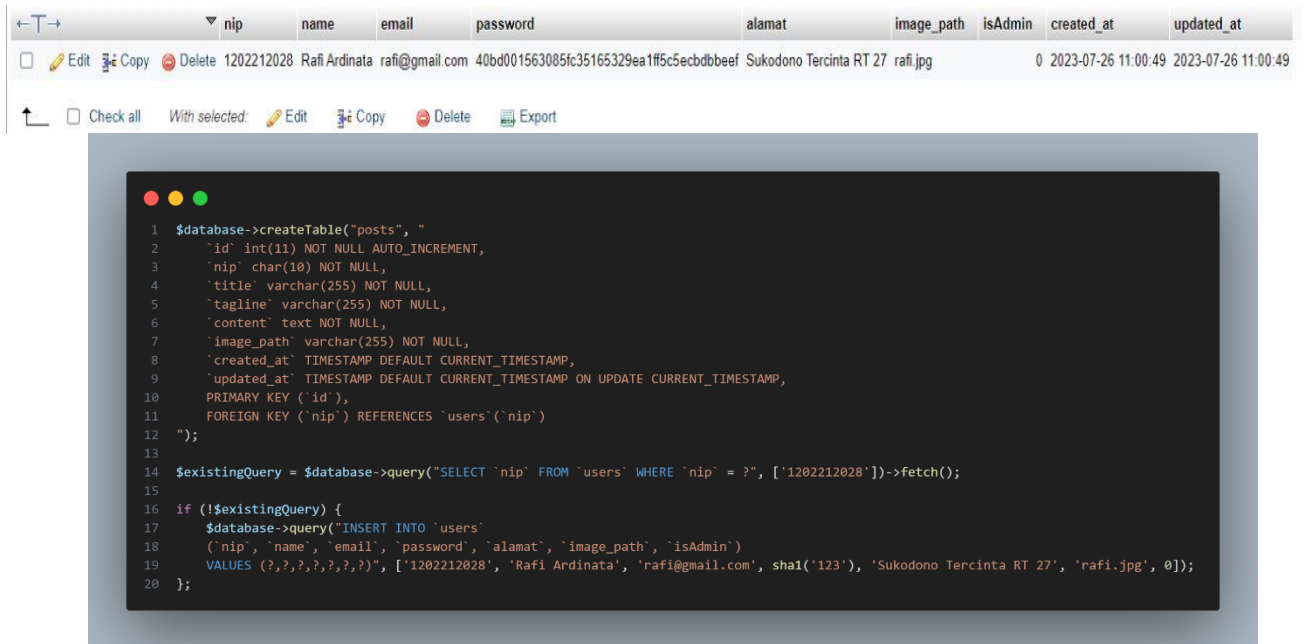
Source code ini bertugas mengelola koneksi dan interaksi dengan database MySQL.

1. Konstruktor (`__construct`):
  - Konstruktor ini akan dipanggil ketika objek dari kelas "Database" dibuat.
  - Konstruktor menerima parameter `$config`, yang berisi konfigurasi untuk menghubungkan ke database, seperti host, username, password, dan nama database.
  - Saat objek "Database" dibuat, koneksi ke database akan segera dilakukan menggunakan informasi dari parameter `$config`.
2. Fungsi `createDB()`:
  - Fungsi ini digunakan untuk membuat database baru jika belum ada.
  - Fungsi ini mengirimkan perintah SQL `CREATE DATABASE IF NOT EXISTS` ke server MySQL untuk membuat database dengan nama yang ada dalam atribut `$dbName`.
3. Fungsi `selectDB()`:
  - Fungsi ini digunakan untuk memilih database yang akan digunakan dalam koneksi.
  - Fungsi ini mengirimkan perintah SQL `USE database_name` ke server MySQL untuk memilih database yang telah dibuat atau sudah ada.
4. Fungsi `createTable($tableName, $query)`:
  - Fungsi ini digunakan untuk membuat tabel baru dalam database.
  - Fungsi ini mengirimkan perintah SQL `CREATE TABLE IF NOT EXISTS` ke server MySQL dengan menggunakan `$tableName` (nama tabel) dan `$query` (definisi struktur tabel) yang diberikan.
5. Fungsi `query($query, $params = [])`:
  - Fungsi ini digunakan untuk mengeksekusi perintah SQL yang kompleks atau menggunakan parameter (prepared statement).
  - Fungsi ini menggunakan prepared statement untuk mencegah serangan SQL injection.
  - Fungsi ini mengembalikan objek "mysqli\_stmt" yang bisa digunakan untuk mendapatkan hasil dari eksekusi perintah.
6. Fungsi `getData($query)`:
  - Fungsi ini digunakan untuk mengambil data dari database berdasarkan perintah SQL yang diberikan melalui `$query`.
  - Fungsi ini mengembalikan hasil dari eksekusi perintah dalam bentuk array asosiatif.
7. Fungsi `getConnection()`:

Fungsi ini digunakan untuk mendapatkan objek koneksi (connection) ke database, yang dapat digunakan dalam interaksi langsung dengan database jika diperlukan.

Setelah mendefinisikan kelas "Database", kode ini kemudian membuat objek "Database" baru dengan menggunakan parameter `$config`. Setelah objek dibuat, maka koneksi ke database akan dilakukan dan database akan dipilih. Selanjutnya, kode ini membuat tabel "users" dalam database jika tabel tersebut belum ada.

Kode ini memiliki peran penting dalam mengatur dan menyediakan koneksi serta interaksi dengan database. Dengan menggunakan kelas "Database" ini, kita dapat dengan mudah membuat dan mengatur koneksi ke database MySQL, serta melakukan operasi SQL seperti membuat tabel atau mengambil data dengan aman menggunakan prepared statement untuk mencegah serangan SQL injection.



Source code ini memiliki beberapa fungsi yang akan dijelaskan sebagai berikut:

1. Fungsi `createTable("posts", $query)`:
  - Fungsi ini digunakan untuk membuat tabel baru dengan nama "posts" dalam database.
  - Tabel "posts" akan memiliki beberapa kolom, seperti ``id``, ``nip``, ``title``, ``tagline``, ``content``, ``image_path``, ``created_at``, dan ``updated_at``.
  - ``id`` adalah kolom dengan tipe data int yang akan digunakan sebagai primary key dan akan secara otomatis bertambah nilainya (AUTO\_INCREMENT).
  - ``nip`` adalah kolom yang akan digunakan sebagai foreign key, mengacu pada kolom ``nip`` di tabel "users".
2. Periksa Data Pengguna (User) yang Sudah Ada:
  - Kode ini menggunakan fungsi `query()` untuk mengambil data dari tabel "users" berdasarkan NIP tertentu ('1202212028' dalam contoh ini).
  - Fungsi `fetch()` digunakan untuk mengambil hasil dari eksekusi query.
  - Jika data dengan NIP yang sama tidak ditemukan (artinya `$existingQuery` bernilai `false`), maka akan dilakukan pendaftaran data pengguna baru dengan NIP '1202212028'.
3. Fungsi `INSERT INTO` untuk Menambah Data Pengguna Baru:
  - Jika data pengguna dengan NIP '1202212028' tidak ditemukan (berdasarkan langkah sebelumnya), maka kode ini akan menggunakan fungsi `query()` untuk memasukkan data pengguna baru ke dalam tabel "users".
  - Data yang dimasukkan meliputi NIP, nama, email, password, alamat, path gambar profil, dan status isAdmin.
  - Password yang dimasukkan di-hash menggunakan fungsi `sha1()`.

Jadi, keseluruhan fungsi kode ini adalah untuk memastikan bahwa ada data pengguna dengan NIP '1202212028' di dalam tabel "users". Jika data tersebut tidak ditemukan, maka data pengguna baru dengan NIP tersebut akan ditambahkan ke dalam tabel "users". Jika data dengan NIP tersebut sudah ada, maka proses penambahan data baru tidak akan dilakukan. Hal ini berguna untuk memastikan bahwa data pengguna dengan NIP yang unik tidak ditambahkan berulang kali ke dalam database.

## C. PENUTUP

### - Saran

Berikut adalah beberapa saran untuk menyelesaikan tugas besar akhir dengan baik:

1. **Perencanaan yang Matang:** Mulailah dengan perencanaan yang baik sebelum mulai mengembangkan sisi admin dan fitur CRUD. Tentukan roadmap proyek secara jelas, identifikasi tugas-tugas yang harus diselesaikan, dan tetapkan tenggat waktu untuk setiap langkah. Pembagian tugas di antara anggota kelompok juga penting untuk memastikan efisiensi dalam pengerjaan proyek.
2. **Pahami Kebutuhan dan Spesifikasi:** Pastikan Anda dan kelompok Anda sepenuhnya memahami kebutuhan dari sisi admin dan fitur CRUD yang akan dikembangkan. Diskusikan dengan baik mengenai apa yang harus diimplementasikan, seperti jenis data yang akan dikelola, bagaimana interaksi admin dengan database, dan bagaimana data akan ditampilkan dalam tampilan yang disesuaikan dengan teman Lexa Institut.
3. **Pelajari Materi yang Dibutuhkan:** Pastikan seluruh anggota kelompok memiliki pemahaman yang baik tentang PHP, Bootstrap, dan konsep CRUD. Jika ada anggota kelompok yang masih perlu memperdalam pemahaman, luangkan waktu untuk mempelajari materi tambahan atau mencari sumber belajar yang relevan.
4. **Gunakan Version Control:** Manfaatkan sistem version control seperti Git untuk mengelola kode proyek. Hal ini akan memudahkan kolaborasi antaranggota kelompok, menghindari kehilangan perubahan, dan memungkinkan kembali ke versi sebelumnya jika diperlukan.
5. **Dokumentasikan Proses:** Selama pengembangan, pastikan untuk mendokumentasikan langkah-langkah yang diambil, masalah yang dihadapi, dan solusi yang diterapkan. Dokumentasi ini akan sangat membantu jika terjadi kendala di masa depan atau ketika ada perluasan proyek di kemudian hari.
6. **Menerima Masukan:** Terima masukan dari dosen atau teman sekelas setelah presentasi. Jika ada saran atau kritik, jadikan itu sebagai kesempatan untuk meningkatkan proyek Anda dan menjadi lebih baik di masa depan.
7. **Beri Komitmen dan Kerja Sama:** Terakhir, tetap berkomitmen dan bekerjalah secara tim dengan baik. Kerjasama yang baik di antara anggota kelompok akan memastikan proyek ini selesai dengan sukses.

Semoga saran-saran di atas membantu kelompok kita menyelesaikan tugas besar akhir dengan sukses dan meraih hasil yang memuaskan. Semoga sukses dalam proyek!

## **- Kesimpulan**

Dapat disimpulkan bahwa kelompok kami telah berhasil menyelesaikan tugas besar akhir UAS dengan baik. Pengembangan situs web tentang pendidikan dengan tema Lexa Institut telah mencakup berbagai aspek penting dari proses pembuatan sebuah website, mulai dari bagian front-end hingga back-end dengan fokus pada implementasi CRUD (Create, Retrieve, Update, Delete) dan sistem login.

Pencapaian kelompok kami dalam menghadapi kendala dan berhasil mengatasinya dengan berdiskusi dan mencari referensi dari berbagai sumber menunjukkan kerjasama yang baik dan semangat dalam menyelesaikan proyek ini. Hal ini juga menunjukkan kemampuan kelompok kami dalam menyelesaikan tantangan yang dihadapi dalam pengembangan web.

Penerapan fitur login yang hanya dapat diakses oleh admin sendiri menunjukkan pemahaman tentang manajemen pengguna yang baik dan juga pentingnya keamanan dalam sebuah situs web. Selain itu, adanya tampilan dashboard admin dengan fitur add user, create article, dan log out menunjukkan kelengkapan fungsionalitas dari sisi admin, yang akan memudahkan admin dalam mengelola konten dan pengguna.

Penerapan fitur CRUD pada halaman create yang memungkinkan pembuatan artikel baru dan kemudian dapat dilihat, diperbarui, dan dihapus menunjukkan fungsionalitas yang lengkap dan interaksi yang baik dengan database. Dengan adanya keterhubungan dengan database, setiap perubahan pada artikel akan langsung ter-update secara real-time, meningkatkan efisiensi dan konsistensi data pada situs web.

Keseluruhan, kelompok kami telah berhasil mencapai tujuan yang telah ditetapkan dalam tugas besar akhir UAS ini dengan baik. Implementasi front-end dan back-end yang dilakukan dengan teknologi seperti HTML, CSS, PHP, dan Bootstrap telah menciptakan sebuah situs web yang lengkap, fungsional, dan dapat digunakan dengan baik. Kerjasama, kreativitas, dan semangat yang ditunjukkan oleh kelompok kami telah menjadi faktor kunci dalam kesuksesan proyek ini.

Dengan selesainya proyek ini, diharapkan situs web Lexa Institut akan menjadi platform pendidikan yang bermanfaat bagi pengguna dan memberikan kemudahan bagi admin dalam mengelola konten. Pengalaman berharga ini juga akan membawa manfaat dalam perjalanan akademik dan profesional kelompok kami di masa depan.