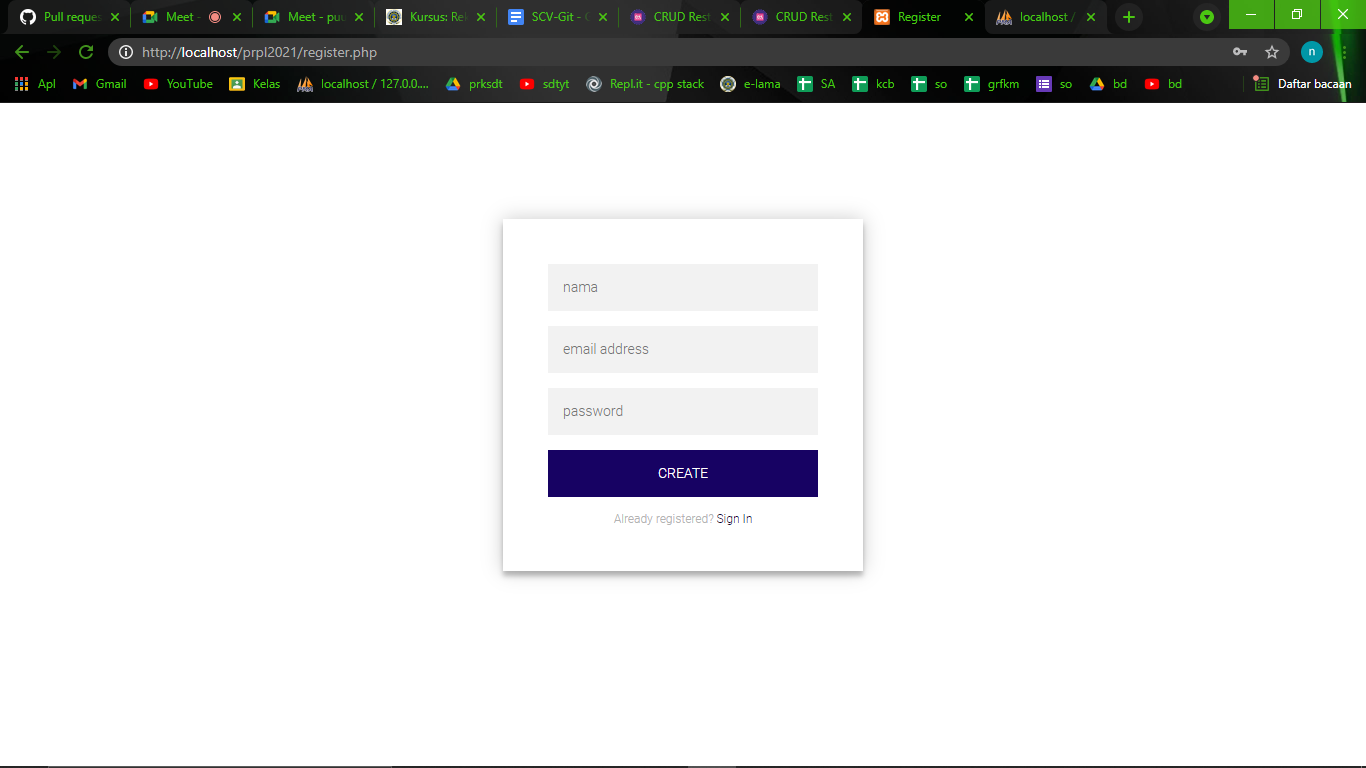
Nama : Nafisah Rahmawati

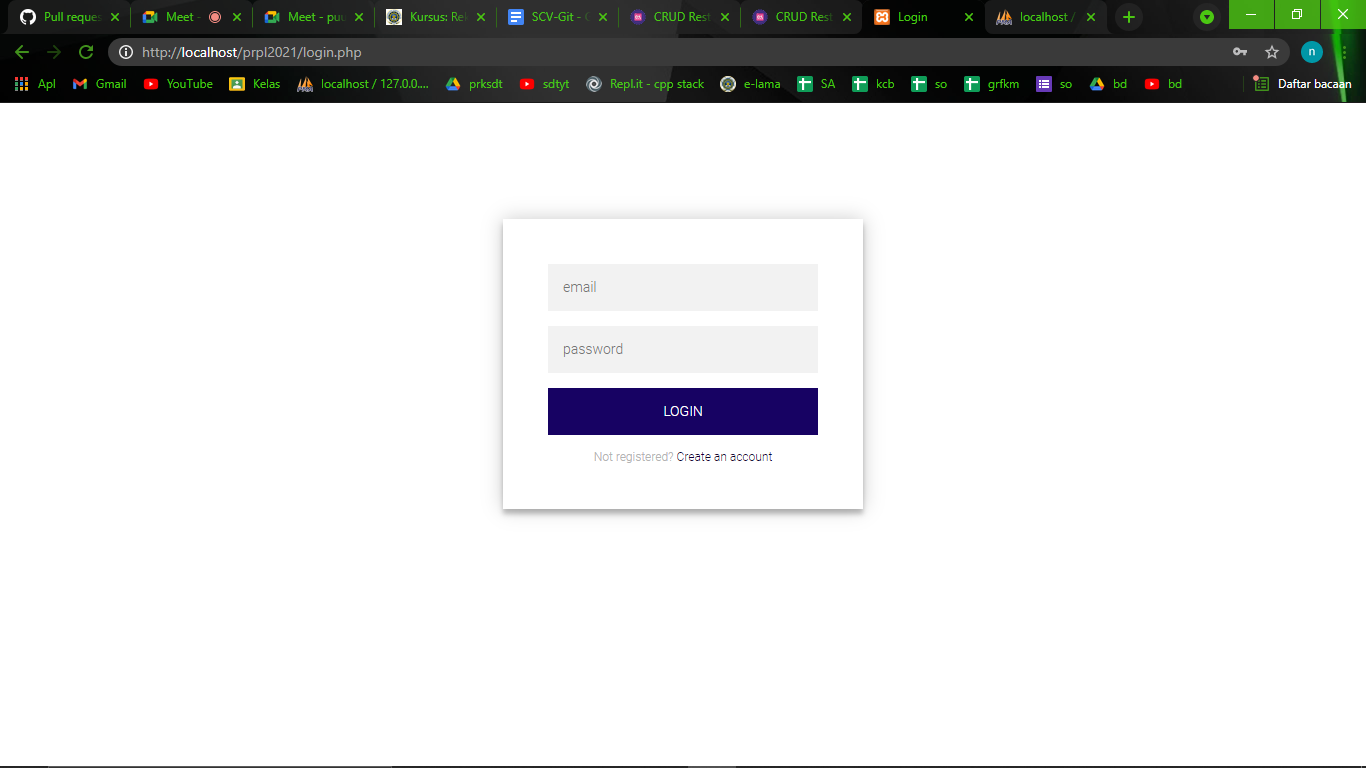
NIM : 1900018213

Kelas : D

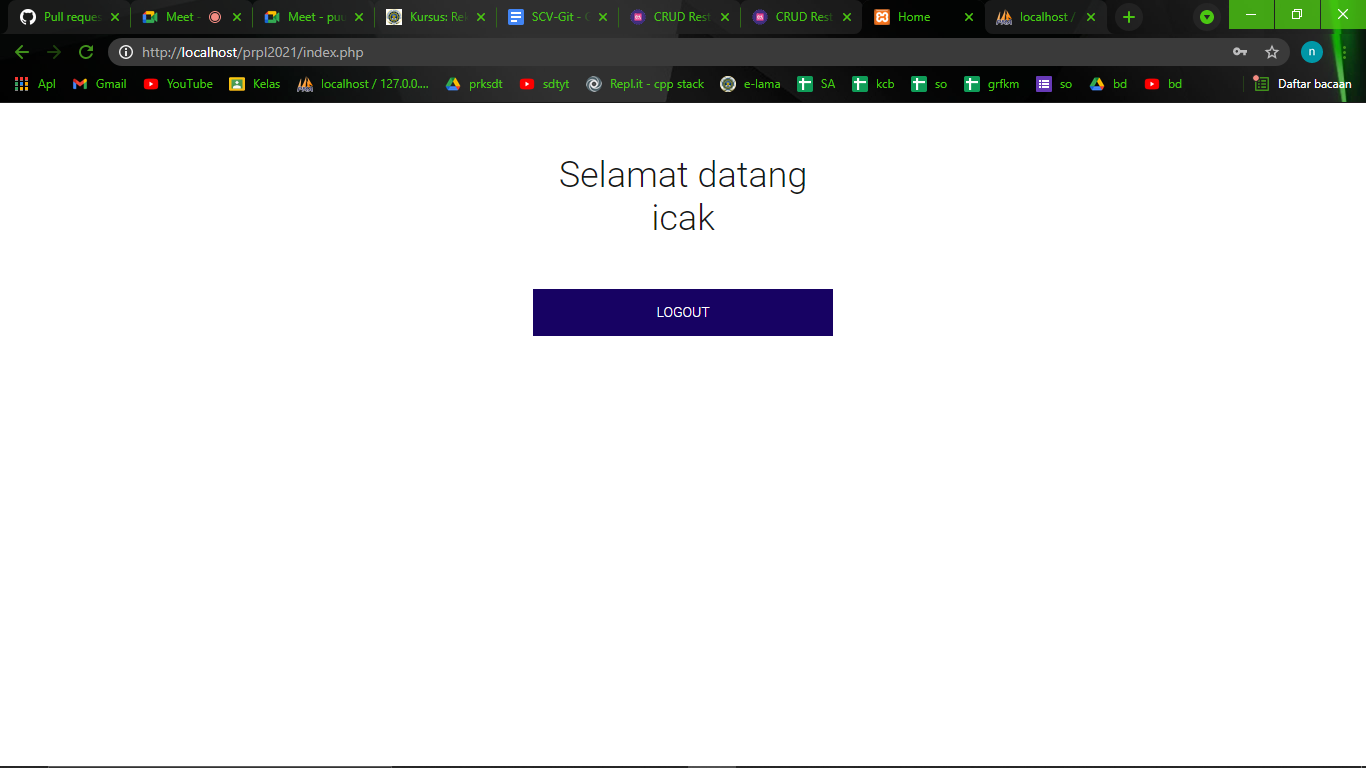
5. analisis kode



Ini adalah tampilan dari register nya



Ini adalah tampilan dari sign in



Ini meruakan tampilan dasbor ketika kita sudah register dan melakukan login

Analisis kode register :

<?php

Ini untuk sambung ke database nya

require\_once "dbconfig.php";

if ($user->isLoggedIn()) {

header("location: index.php");

}

if (isset($\_POST['kirim'])) {

Ini berfungsi untuk menampung data ke database nya agar tersimpan dan bisa dibuka kapan aja jika mau

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

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

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

if (strlen($password) < 8 || strlen($password) > 12) {

$error = true;

Sc ini untuk sambung ke register.php

// return header("location: register.php");

} else {

if ($user->register($nama, $email, $password)) {

$user->sender($email);

$success = true;

} else {

$error = true;

}

}

// if ($user->sender($email)) {

// echo "berhasil";

// }

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Register</title>

<link rel="stylesheet" href="style.css" media="screen" title="no title" charset="utf-8">

</head>

<body>

<div class="login-page">

<div class="form">

<form class="register-form" method="post">

<?php if (isset($error)) : ?>

<div class="error">

Email Sudah digunakan atau password salah

</div>

<?php endif; ?>

<?php if (isset($success)) : ?>

<div class="success">

Berhasil mendaftar. Silakan <a href="login.php">login</a>.

</div>

<?php endif; ?>

<input type="text" name="nama" placeholder="nama" required />

<input type="email" name="email" placeholder="email address" required />

<input type="password" name="password" placeholder="password" required />

<button type="submit" name="kirim">create</button>

<p class="message">Already registered? <a href="login.php">Sign In</a></p>

</form>

</div>

</div>

</body>

</html>

Analisis kode login

<?php

require\_once "dbconfig.php";

if ($user->isLoggedIn()) {

header("location: index.php");

}

if (isset($\_POST['kirim'])) {

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

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

if ($user->login($email, $password)) {

header("location: index.php");

} else {

$error = $user->getLastError(); }

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Login</title>

<link rel="stylesheet" href="style.css" media="screen" title="no title" charset="utf-8">

</head>

<body>

<div class="login-page">

<div class="form">

Fungsi method = post itu untuk mengambil data dari form

<form class="login-form" method="post">

<?php if (isset($error)) : ?>

<div class="error">

<?php echo $error ?>

</div>

<?php endif; ?>

<input type="email" name="email" placeholder="email" required />

<input type="password" name="password" placeholder="password" required />

<button type="submit" name="kirim">login</button>

<p class="message">Not registered? <a href="register.php">Create an account</a></p>

</form>

</div>

</div>

</body>

</html>

Analisis kode index

Index ini merupakan halaman utama dari sistem login yang tadi

<?php

// Lampirkan dbconfig

require\_once "dbconfig.php";

// Cek status login user

if (!$user->isLoggedIn()) {

header("location: login.php"); //Redirect ke halaman login

}

// Ambil data user saat ini

$currentUser = $user->getUser();

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Home</title>

<link rel="stylesheet" href="style.css" media="screen" title="no title" charset="utf-8">

</head>

<body>

<div class="container">

<div class="info">

<h1>Selamat datang <?php echo $currentUser['nama'] ?></h1>

</div>

<a href="logout.php"><button type="button">Logout</button></a>

</div>

</body>

</html>

Analisis kode AuthClass :

{

private $db;

private $error;

function \_\_construct($db\_conn)

{

$this->db = $db\_conn;

session\_start();

}

public function register($nama, $email, $password)

{

try {

Agar password terenkripsi

$hashPasswd = password\_hash($password, PASSWORD\_DEFAULT);

$stmt = $this->db->prepare("INSERT INTO tbAuth(nama, email, password) VALUES(:nama, :email, :pass)");

$stmt->bindParam(":nama", $nama);

Sebagai penampung variabel

$stmt->bindParam(":email", $email);

$stmt->bindParam(":pass", $hashPasswd);

$stmt->execute();

return true;

Sebagai alert bahwa email yang sudah daftar tidak bisa diapakai untuk daftar kedua kali

} catch (PDOException $e) {

if ($e->errorInfo[0] == 23000) {

$this->error = "Email sudah digunakan!";

return false;

} else {

echo $e->getMessage();

return false;

}

}

}

public function login($email, $password)

{

try {

$stmt = $this->db->prepare("SELECT \* FROM tbAuth WHERE email = :email");

$stmt->bindParam(":email", $email);

$stmt->execute();

$data = $stmt->fetch();

if ($stmt->rowCount() > 0) {

if (password\_verify($password, $data['password'])) {

Sebagai alert bahwa yang password tidak sesuai dengan yang diregister

$\_SESSION['user\_session'] = $data['id'];

return true;

} else {

$this->error = "Email atau Password Salah";

return false;

}

} else {

$this->error = "Email atau Password Salah";

return false;

}

} catch (PDOException $e) {

echo $e->getMessage();

return false;

}

}

public function isLoggedIn()

{

if (isset($\_SESSION['user\_session'])) {

return true;

}

}

public function getUser()

{

if (!$this->isLoggedIn()) {

return false;

}

try {

$stmt = $this->db->prepare("SELECT \* FROM tbAuth WHERE id = :id");

$stmt->bindParam(":id", $\_SESSION['user\_session']);

$stmt->execute();

return $stmt->fetch();

} catch (PDOException $e) {

echo $e->getMessage();

return false;

}

}

public function logout()

{

session\_destroy();

unset($\_SESSION['user\_session']);

return true;

}

public function getLastError()

{

return $this->error;

}

public function sender($email)

{

$to = $email; // Send email to our user

$subject = 'Signup | Verification'; // Give the email a subject

$message = '

Your account is successfully created

Fungsinya untuk mendapatkan pesan dari email

Please enjoy

Nafisah Rahmawati';

mail($to, $subject, $message); // Send our email

}

}

Kesimpulan

Pada proses sign in saya menggunakan php native dan function. Salah satu nya saya menggunakan AuthClass sebagai implementasi OOP. Proses sign up saya menggunakan method POST dan kemudian data itu disimpan kedalam database, database yang saya gunakan disini phpmyadmin MySQLi. Jika kita sudah register menggunakan email, maka email tersebut sudah bisa di pakai buat login. Dan email tersebut tidak bisa digunakan berkali-kali. Ketika email sudah terdaftar maka aktivasi di email nya akan masuk sebagai pemberitahuan. Dan untuk sign in nya kita sudah jika masuk maka akan dialihkan ke dasbor.