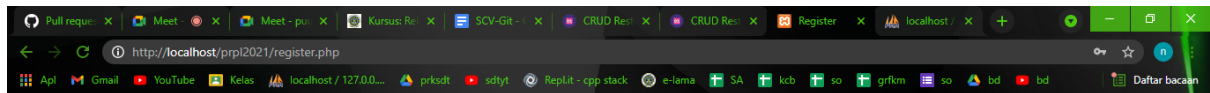


Nama : Nafisah Rahmawati

NIM : 1900018213

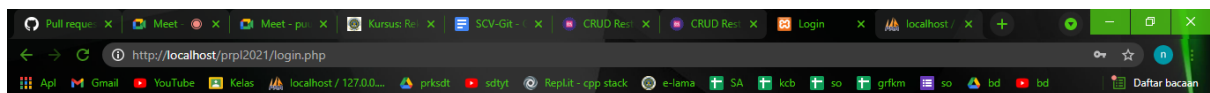
Kelas : D

5. analisis kode



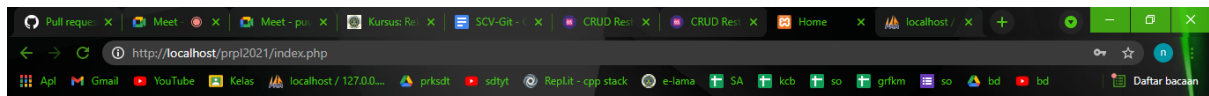
[Already registered? Sign In](#)

Ini adalah tampilan dari register nya



[Not registered? Create an account](#)

Ini adalah tampilan dari sign in



Selamat datang
icak

LOGOUT

Ini meruakan tampilan dasbor ketika kita sudah register dan melakukan login

Analisis kode register :

<?php

```
require_once "dbconfig.php";  
if ($user->isLoggedIn()) {  
    header("location: index.php");  
}
```

Ini untuk sambung ke database nya

```
if (isset($_POST['irim'])) {  
    $nama = $_POST['nama'];  
    $email = $_POST['email'];  
    $password = $_POST['password'];
```

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

```
if (strlen($password) < 8 || strlen($password) > 12) {  
    $error = true;  
    // return header("location: register.php");  
} else {  
    if ($user->register($nama, $email, $password)) {  
        $user->sender($email);  
        $success = true;  
    } else {  
        $error = true;  
    }  
}
```

Sc ini untuk sambung ke register.php

```

    // 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 = $user->getLastError(); }
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Login</title>
    <link rel="stylesheet" href="style.css" media="screen" title="no title" charset="utf-8">
</head>
<body>
    <div class="login-page">
        <div class="form">
            <form class="login-form" method="post">
                <?php if (isset($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>

```

Fungsi method = post itu untuk mengambil data dari form

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 {
        $hashPasswd = password_hash($password, PASSWORD_DEFAULT); → Agar password terenkripsi

        $stmt = $this->db->prepare("INSERT INTO tbAuth(nama, email, password) VALUES(:nama,
:email, :pass)");
        $stmt->bindParam(":nama", $nama);
        $stmt->bindParam(":email", $email); → Sebagai penampung variabel
        $stmt->bindParam(":pass", $hashPasswd);
        $stmt->execute();
        return true;
    } catch (PDOException $e) {
        if ($e->errorInfo[0] == 23000) { → Sebagai alert bahwa email yang sudah
            $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'])) {
                $_SESSION['user_session'] = $data['id'];
                return true; → Sebagai alert bahwa yang
            } 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

Please enjoy

Nafisah Rahmawati';

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



Fungsinya untuk mendapatkan
pesan dari 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.