**Tutorial PHP & MySQL: Membuat Login dan Register (dengan Bootstrap 4)**

[**#PHP**](https://www.petanikode.com/topik/php) [**#MySQL**](https://www.petanikode.com/topik/mysql)

Login dan register adalah fitur yang banyak dibutuhkan dalam pengembangan aplikasi web.

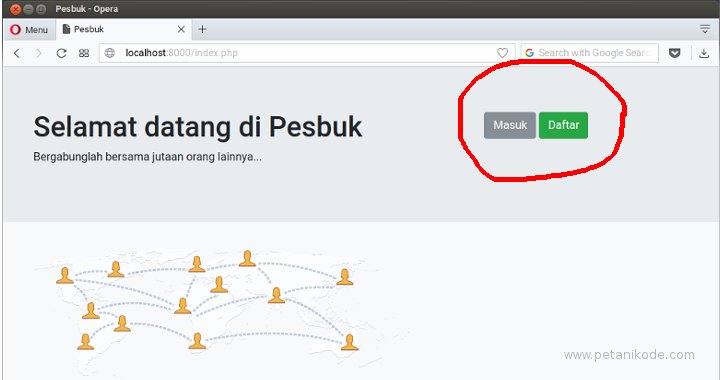
Biasanya saat membuat halaman admin. Kita pasti akan membuat login dan register.

Lalu bagaimana cara membuatnya?

Tulisan ini akan menjawabnya…

Sebelum itu, saya ingin menyampaikan apa saja yang akan dibahas di sini.

* Tutorial ini menggunakan PHP Native dan PDO sebagai koneksi ke database;
* Tutorial ini menggunakan [Framework Bootstrap 4](https://www.petanikode.com/bootstrap4/) untuk tampilan;
* Tutorial ini hanya akan fokus membahas fitur login dan register saja, untuk class-class bootstrap saya tidak akan jelaskan. Saya anggap, pembaca sudah paham dengan Bootstrap.



Baiklah, mari kita mulai…

**Persiapan Alat dan Bahan**

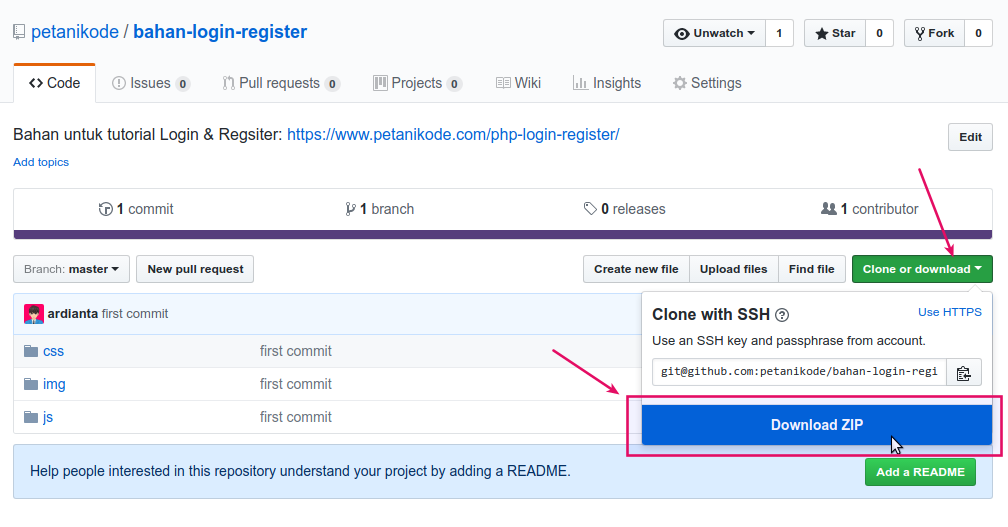
Berikut ini alat-alat yang dibutuhkan:

1. **Teks Editor**: [VS Code](https://www.petanikode.com/text-editor-vscode/) (pakai yang lain juga boleh);
2. **Web Browser**: Google Chrome, Opera, Firefox, dll;
3. **Server**: PHP (versi 5.6 ke atas), Apache2/Nginx, dan MySQL.

Sementara untuk bahan-bahannya silahkan sediakan:

1. [Bootstrap 4](https://www.petanikode.com/bootstrap4);
2. Beberapa gambar.

Silahkan download di sini semua bahannya: [bahan-bahan.zip](https://github.com/petanikode/bahan-login-register)



Alat dan bahan sudah siap.

Mari kita mulai  *coding*… 😄

**Membuat Project Baru**

Buatlah sebuah direktori baru bernama php-login-register di htdocs buat yang menggunakan XAMPP. Sedangkan yang menggunakan server **apache2** dan **nginx**, silahkan buat di /var/www/html.

Ikuti perintah berikut:

sudo chmod 777 /var/www/html -R # berikan hak akses dulu

mkdir /var/www/html/php-login-register # buat direktori baru

*cd* /var/www/html/php-login-register # pindah ke direktori baru

Buat tiga direktori di dalamnya:

mkdir css js img # buat tiga direktori sekaligus

Setelah itu buat file yang dibutuhkan:

# membuat file kosong

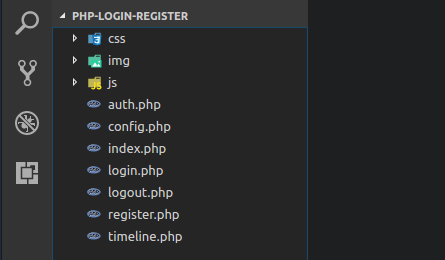
touch index.php login.php register.php logout.php timeline.php auth.php config.php

Buka dengan VS Code:

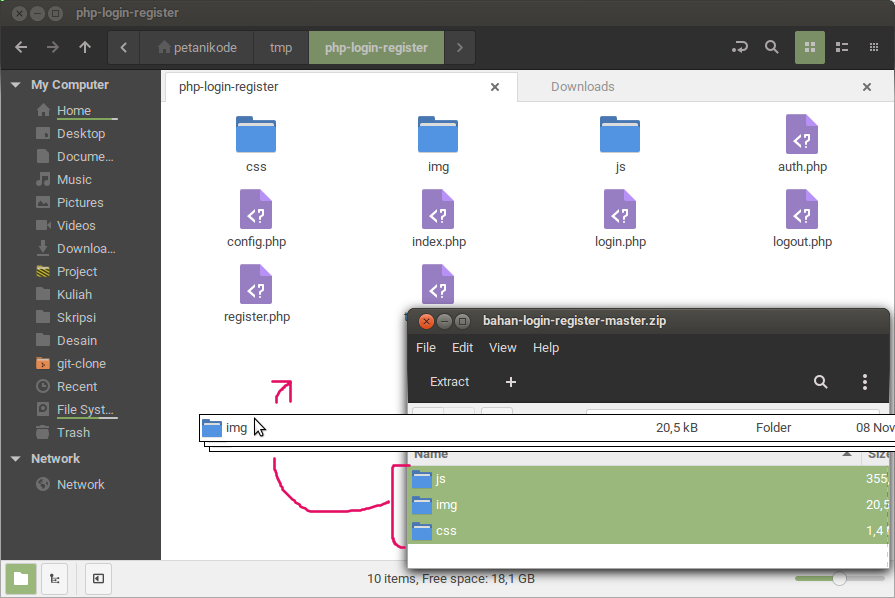
# buka direktori ini dengan VS Code, pastikan kamu sudah menginstall VS Code

code .

Sehingga kita memiliki struktur direktori seperti ini:



Setelah itu, silahkan ekstrak dan *copy* semua bahan yang di-download tadi ke sana.



Beres…

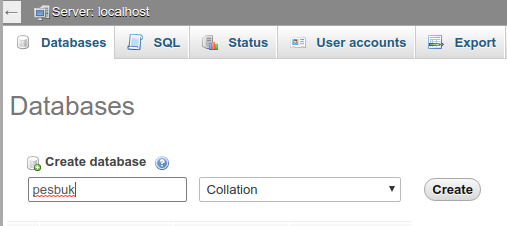
Semua bahan-bahan sudah siap.

**Menyiapkan Database**

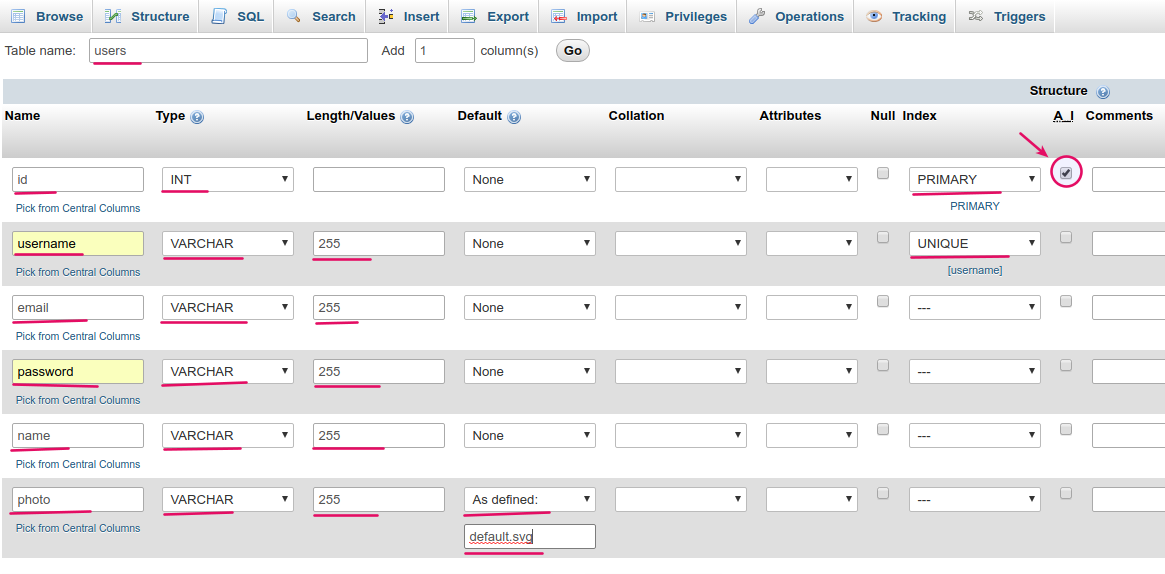
Sebelum kita mulai menulis kode, siapkan dulu database-nya.

Silahkan buka phpmyadmin http://localhost/phpmyadmin/ dan buat database baru bernama pesbuk.

Baca juga: [Cara Masuk ke phpMyAdmin Tanpa Login cPanel](https://www.niagahoster.co.id/blog/cara-masuk-ke-phpmyadmin/).



Setelah itu, buatlah sebuah tabel baru dengan nama users dan buat kolomnya seperti ini:



Kode SQL-nya:

CREATE TABLE `users` (

`id` *int*(11) NOT NULL,

`username` *varchar*(255) NOT NULL,

`email` *varchar*(255) NOT NULL,

`password` *varchar*(255) NOT NULL,

`name` *varchar*(255) NOT NULL,

`photo` *varchar*(255) NOT NULL DEFAULT 'default.svg'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Indexes for table `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `username` (`username`);

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

MODIFY `id` *int*(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

Selesai, selanjutnya kita mulai *coding*…

**Membuat Koneksi ke Database**

Hal pertama yang harus kita lakukan adalah membuat kode untuk koneksi ke database.

Silahkan buka file config.php, lalu isi kodenya seperti ini:

<?php

*$db\_host* = "localhost";

*$db\_user* = "root";

*$db\_pass* = "kopi";

*$db\_name* = "pesbuk";

try {

//create PDO connection

*$db* = new PDO("mysql:host=$db\_host;dbname=$db\_name", *$db\_user*, *$db\_pass*);

} catch(PDOException *$e*) {

//show error

die("Terjadi masalah: " . *$e*->getMessage());

}

Sesuaikan **user** dan **password**-nya dengan konfigurasi di komputermu. Karena di komputer saya, **password** mysql-nya menggunakan kopi.

Jika tidak menggunakan password, maka dikosongkan saja.

*$db\_pass* = "";

**Apa maksud kode config.php di atas?**

Pertama kita membuat variabel untuk menyimpan konfigurasi database:

*$db\_host* = "localhost";

*$db\_user* = "root";

*$db\_pass* = "kopi";

*$db\_name* = "pesbuk";

Setelah itu, kita membuat sebuah objek baru bernama $db, objek ini kita perlukan untuk melakukan query ke database.

*$db* = new PDO("mysql:host=$db\_host;dbname=$db\_name", *$db\_user*, *$db\_pass*);

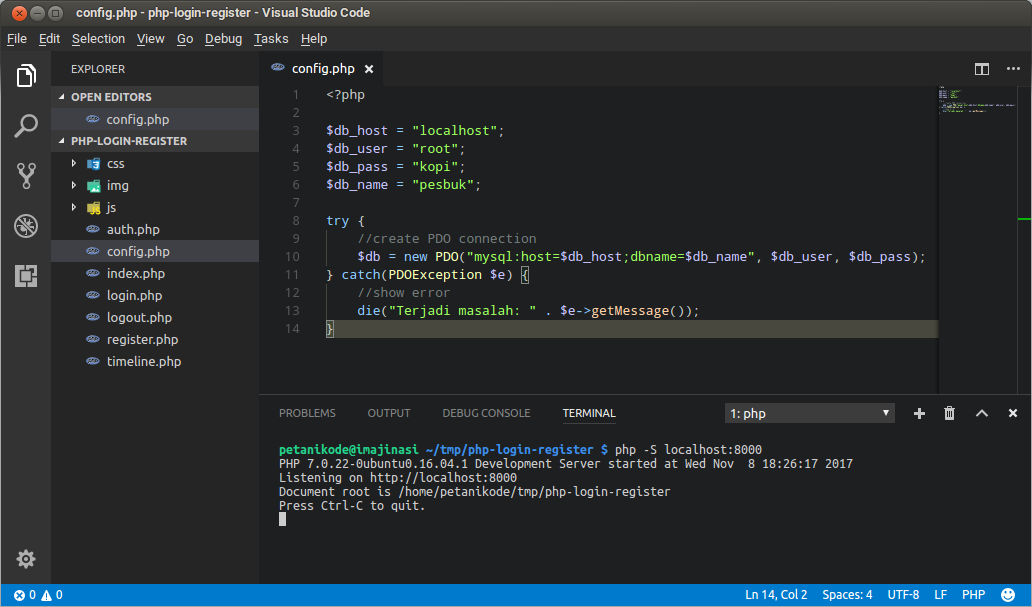
Saat kita membuat objek PDO, kita gunakan blok *Try/Catch* untuk menangai error.

Coba kita tes.

Silahkan buka terminal di VS Code (Tekan Ctrl+`), lalu jalankan server PHP di sana dengan perintah:

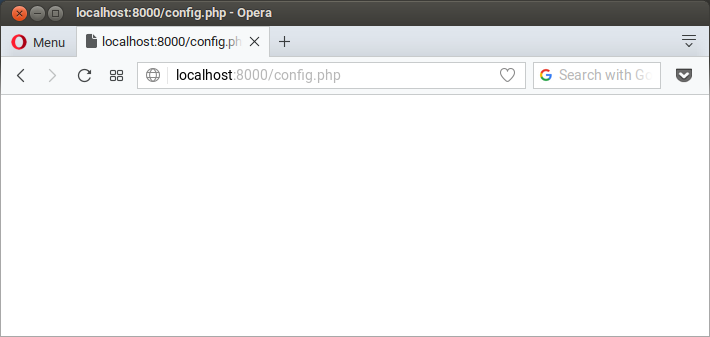
php -S localhost:8000

Argumen -S menggunakan huruf besar/kapital.

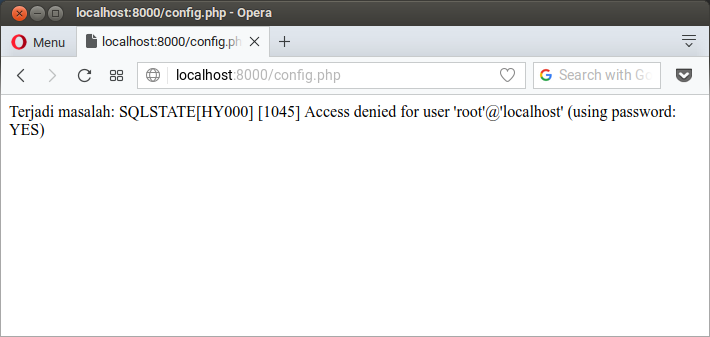


Sekarang buka: http://localhost:8000/config.php

Jika tampilannya kosong, maka koneksi berhasil dan tidak ada error.



Tapi kalau ada error, biasanya akan tampil seperti ini.



**Membuat Halaman Utama**

Halaman utama adalah halaman yang akan dibuka pertama kali oleh pengunjung web.

Halaman ini bertugas sebagai *landing page*, isinya cuma html saja.

Silahkan buka file index.php lalu isi kodenya seperti ini:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Pesbuk</title>

<!-- menyisipkan bootstrap -->

<link rel="stylesheet" href="css/bootstrap.min.css" />

</head>

<body class="bg-light">

<header>

<div class="jumbotron jumbotron-fluid">

<div class="container">

<div class="row">

<div class="col-md-8">

<h1>Selamat datang di Pesbuk</h1>

<p>Bergabunglah bersama jutaan orang lainnya...</p>

</div>

<div class="col-md-4">

<a href="login.php" class="btn btn-secondary">Masuk</a>

<a href="register.php" class="btn btn-success">Daftar</a>

</div>

</div>

</div>

</div>

</header>

<section>

<div class="container">

<div class="row">

<div class="col-md-12">

<img class="img img-responsive" src="img/connect.png" />

</div>

</div>

</div>

</section>

</body>

</html>

Setelah itu, coba buka http://localhost:8000/.



Keren ‘kan?

O ya gambarnya kita pinjam dari Facebook 😄.

Tidak apa-apa, ini kan hanya untuk belajar saja. Nanti kalau sudah buat web sungguhan, **tidak boleh** menggunakan gambar orang lain **tanpa izin**.

**Membuat Fitur Register di PHP**

Berikutnya kita akan membuat fitur registrasi.

Alur kerjanya:

1. Input data dari form;
2. Filter data yang diinputkan;
3. Simpan ke database;
4. Kalau berhasil, alihkan ke halaman login.

Mari kita buat.

Silahkan buka file register.php, lalu isi dengan kode berikut:

<?php

require\_once("config.php");

if(isset(*$\_POST*['register'])){

// filter data yang diinputkan

*$name* = filter\_input(INPUT\_POST, 'name', FILTER\_SANITIZE\_STRING);

*$username* = filter\_input(INPUT\_POST, 'username', FILTER\_SANITIZE\_STRING);

// enkripsi password

*$password* = password\_hash(*$\_POST*["password"], PASSWORD\_DEFAULT);

*$email* = filter\_input(INPUT\_POST, 'email', FILTER\_VALIDATE\_EMAIL);

// menyiapkan query

*$sql* = "INSERT INTO users (name, username, email, password)

VALUES (:name, :username, :email, :password)";

*$stmt* = *$db*->prepare(*$sql*);

// bind parameter ke query

*$params* = array(

":name" => *$name*,

":username" => *$username*,

":password" => *$password*,

":email" => *$email*

);

// eksekusi query untuk menyimpan ke database

*$saved* = *$stmt*->execute(*$params*);

// jika query simpan berhasil, maka user sudah terdaftar

// maka alihkan ke halaman login

if(*$saved*) header("Location: login.php");

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Register Pesbuk</title>

<link rel="stylesheet" href="css/bootstrap.min.css" />

</head>

<body class="bg-light">

<div class="container mt-5">

<div class="row">

<div class="col-md-6">

<p>&larr; <a href="index.php">Home</a>

<h4>Bergabunglah bersama ribuan orang lainnya...</h4>

<p>Sudah punya akun? <a href="login.php">Login di sini</a></p>

<form action="" method="POST">

<div class="form-group">

<label for="name">Nama Lengkap</label>

<input class="form-control" type="text" name="name" placeholder="Nama kamu" />

</div>

<div class="form-group">

<label for="username">Username</label>

<input class="form-control" type="text" name="username" placeholder="Username" />

</div>

<div class="form-group">

<label for="email">Email</label>

<input class="form-control" type="email" name="email" placeholder="Alamat Email" />

</div>

<div class="form-group">

<label for="password">Password</label>

<input class="form-control" type="password" name="password" placeholder="Password" />

</div>

<input type="submit" class="btn btn-success btn-block" name="register" value="Daftar" />

</form>

</div>

<div class="col-md-6">

<img class="img img-responsive" src="img/connect.png" />

</div>

</div>

</div>

</body>

</html>

**Penjelasan:**

Pertama kita membutuhkan objek $db untuk melakukan query, maka kita impor config.php dengan fungsi require\_once().

require\_once("config.php");

Setelah itu, baru kita buat kode untuk menyimpan data ke database.

Sebelumnya, kita cek dulu. Apakah tombol **Register** sudah ditekan atau belum:

if(isset(*$\_POST*['register'])){

// ...

}

?>

Selanjutnya melakukan filter dan enkripsi password terhadap data yang diinputkan:

// filter data yang diinputkan

*$name* = filter\_input(INPUT\_POST, 'name', FILTER\_SANITIZE\_STRING);

*$username* = filter\_input(INPUT\_POST, 'username', FILTER\_SANITIZE\_STRING);

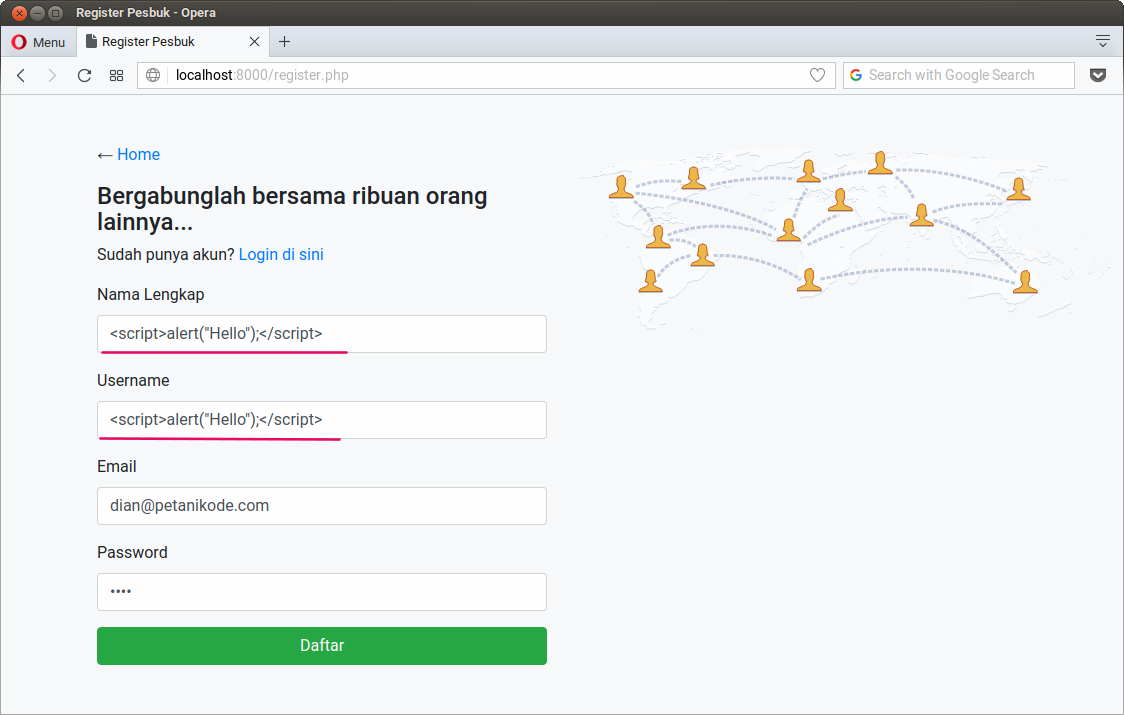
// enkripsi password

*$password* = password\_hash(*$\_POST*["password"], PASSWORD\_DEFAULT);

*$email* = filter\_input(INPUT\_POST, 'email', FILTER\_VALIDATE\_EMAIL);

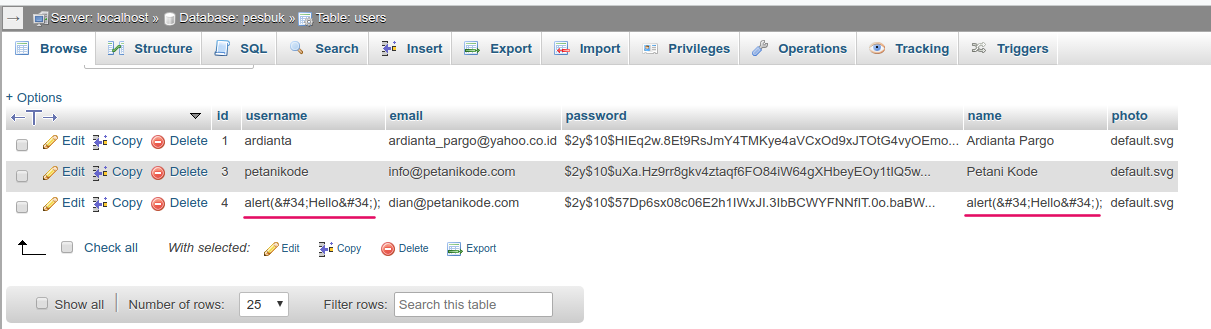
Kenapa harus di-filter dan dienkripsi?

Jangan percaya pada apa yang diinputkan user, bisa saja mereka menginputkan username dan name seperti ini:



Ini salah satu bentuk [serangan XSS](https://www.petanikode.com/codeigniter-serangan-xss/).

Karena itu, kita harus memfilternya agar skrip yang diinputkan user diubah menjadi karakter yang aman.



Lalu kenapa password harus dienkripsi?

Tujuannya untuk keamanan. Nanti kalau database dicuri, si pencuri tidak akan tahu password dari akun-akun yang sudah terdaftar.

// enkripsi password

*$password* = password\_hash(*$\_POST*["password"], PASSWORD\_DEFAULT);

Tunggu dulu…

Kenapa password tidak di-filter?

Karena agar user dapat menggunakan simbol untuk membuat password.

Kalau kita filter nanti user tidak bisa membuat password dengan simbol-simbol seperti: &, <, >, %, dll.

Setelah data di-filter, selanjutnya kita buat query dengan objek $db.

*$sql* = "INSERT INTO users (name, username, email, password)

VALUES (:name, :username, :email, :password)";

*$stmt* = *$db*->prepare(*$sql*);

:name, :username, :email, :password adalah *placeholder*. Tujuannya agar terhindar dari serangan **SQL Injection**.

Setelah itu, kita *bind* parameter atau mengisi *placeholder* dengan nilai yang kita dapatkan dari hasil filter.

// bind parameter ke query

*$params* = array(

":name" => *$name*,

":username" => *$username*,

":password" => *$password*,

":email" => *$email*

);

Selanjutnya, kita eksekusi query-nya:

// eksekusi query untuk menyimpan ke database

*$saved* = *$stmt*->execute(*$params*);

Eksekusi query akan menghasilkan nilai true jika berhasil.

Sehingga kita bisa membuat sebuah kondisi:

// jika query simpan berhasil, maka user sudah terdaftar

// maka alihkan ke halaman login

if(*$saved*) header("Location: login.php");

Lalu yang terakhir adalah kode untuk form registernya:

<form action="" method="POST">

...

</form>

Atribut action="" artinya data akan dikirim ke file itu sendiri, yaitu: register.php.

Mudah bukan?

Selanjutnya kita akan membuat fitur login.

**Membuat Fitur Login di PHP**

Silahkan buka file login.php lalu isi—ketik! jangan copas—dengan kode berikut:

<?php

require\_once("config.php");

if(isset(*$\_POST*['login'])){

*$username* = filter\_input(INPUT\_POST, 'username', FILTER\_SANITIZE\_STRING);

*$password* = filter\_input(INPUT\_POST, 'password', FILTER\_SANITIZE\_STRING);

*$sql* = "SELECT \* FROM users WHERE username=:username OR email=:email";

*$stmt* = *$db*->prepare(*$sql*);

// bind parameter ke query

*$params* = array(

":username" => *$username*,

":email" => *$username*

);

*$stmt*->execute(*$params*);

*$user* = *$stmt*->fetch(PDO::FETCH\_ASSOC);

// jika user terdaftar

if(*$user*){

// verifikasi password

if(password\_verify(*$password*, *$user*["password"])){

// buat Session

session\_start();

*$\_SESSION*["user"] = *$user*;

// login sukses, alihkan ke halaman timeline

header("Location: timeline.php");

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Login Pesbuk</title>

<link rel="stylesheet" href="css/bootstrap.min.css" />

</head>

<body class="bg-light">

<div class="container mt-5">

<div class="row">

<div class="col-md-6">

<p>&larr; <a href="index.php">Home</a>

<h4>Masuk ke Pesbuk</h4>

<p>Belum punya akun? <a href="register.php">Daftar di sini</a></p>

<form action="" method="POST">

<div class="form-group">

<label for="username">Username</label>

<input class="form-control" type="text" name="username" placeholder="Username atau email" />

</div>

<div class="form-group">

<label for="password">Password</label>

<input class="form-control" type="password" name="password" placeholder="Password" />

</div>

<input type="submit" class="btn btn-success btn-block" name="login" value="Masuk" />

</form>

</div>

<div class="col-md-6">

<!-- isi dengan sesuatu di sini -->

</div>

</div>

</div>

</body>

</html>

**Penjelasan:**

Algoritmanya hampir sama seperti register.

Pertama kita mengimpor file config.php:

require\_once("config.php");

Setelah itu, kita cek apakah tombol login sudah ditekan atau belum:

if(isset(*$\_POST*['login'])){

// ...

}

Lalu melakukan filter:

*$username* = filter\_input(INPUT\_POST, 'username', FILTER\_SANITIZE\_STRING);

*$password* = filter\_input(INPUT\_POST, 'password', FILTER\_SANITIZE\_STRING);

Selanjutnya melakukan query:

*$sql* = "SELECT \* FROM users WHERE username=:username OR email=:email";

*$stmt* = *$db*->prepare(*$sql*);

Pada query, kita menggunakan logika OR agar user bisa login dengan email dan username .

Selanjutnya *bind* parameter dan ekekusi querynya.

// bind parameter ke query

*$params* = array(

":username" => *$username*,

":email" => *$username*

);

*$stmt*->execute(*$params*);

Setelah query dieksekusi, kita ambil hasilnya dengan fungsi fetch().

*$user* = *$stmt*->fetch(PDO::FETCH\_ASSOC);

Artinya, kita akan mengambil hasil query sebagai array assosiatif.

Fungsi fetch() akan menghasilkan null kalau tidak ada data, maka kita bisa membuat kondisi seperti ini:

// jika user terdaftar

if(*$user*){

//...

}

Selanjutnya, kita lakukan verifikasi password yang diinputkan oleh user dengan password yang ada di database.

if(password\_verify(*$password*, *$user*["password"])){

//...

}

Fungsi password\_verify() akan menghasilkan nilai true kalau password yang diinputkan sama dengan yang di database.

Langkah terakhir adalah membuat *session* dan mengalihkannya ke halaman timeline.php.

// buat Session

session\_start();

*$\_SESSION*["user"] = *$user*;

// login sukses, alihkan ke halaman timeline

header("Location: timeline.php");

Variabel $\_SESSION adalah variabel super global yang bisa dibaca dari semua file PHP, karena diingat oleh server.

Untuk dapat menggunakan variabel $\_SESSION kita harus memanggil fungsi session\_start() terlebih dahulu.

$\_SESSION["user"] = $user; artinya kita menyimpan data user ke dalam *session*.

Ini yang akan kita manfaatkan untuk mengecek apakah user sudah login atau belum.

**Membuat Halaman Timeline**

Sebelum membuat kode untuk halaman timeline.php, kita buat dulu auth.php.

File auth.php berfungsi untuk mengecek *session*, apakah user sudah login atau belum.

Silahkan buka file auth.php kemudian isi seperti ini:

<?php

session\_start();

if(!isset(*$\_SESSION*["user"])) header("Location: login.php");

**Penjelasan:**

Seperti penjelasan sebelumnya, saat kita ingin menggunakan variabel $\_SESSION, maka kita harus memanggil fungsi session\_start() terlebih dahulu.

Setelah itu kita cek dengan:

if(!isset(*$\_SESSION*["user"])) header("Location: login.php");

Artinya, jika variabel $\_SESSION["user"] tidak memiliki nilai, maka user belum login… paksa dia untuk login 😄.

Tanda seru (!) di depan isset() artinya tidak. Berarti tidak ter-set (variabel belum dibuat).

Nah barulah sekarang kita buat halaman timeline.php.

Silahkan buka file timeline.php, lalu isi dengan kode berikut:

<?php require\_once("auth.php"); ?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Pesbuk Timeline</title>

<link rel="stylesheet" href="css/bootstrap.min.css" />

</head>

<body class="bg-light">

<div class="container mt-5">

<div class="row">

<div class="col-md-4">

<div class="card">

<div class="card-body text-center">

<img class="img img-responsive rounded-circle mb-3" width="160" src="img/<?php echo $\_SESSION['user']['photo'] ?>" />

<h3><?php echo $\_SESSION["user"]["name"] ?></h3>

<p><?php echo $\_SESSION["user"]["email"] ?></p>

<p><a href="logout.php">Logout</a></p>

</div>

</div>

</div>

<div class="col-md-8">

<form action="" method="post" />

<div class="form-group">

<textarea class="form-control" placeholder="Apa yang kamu pikirkan?"></textarea>

</div>

</form>

<?php for($i=0; $i < 6; $i++){ ?>

<div class="card mb-3">

<div class="card-body">

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis veritatis nemo ad recusandae labore nihil iure qui eum consequatur, officiis facere quis sunt tempora impedit ullam reprehenderit facilis ex amet!

</div>

</div>

<?php } ?>

</div>

</div>

</div>

</body>

</html>

Halaman timeline.php hanya boleh dibuka oleh user yang sudah login saja. Maka kita harus mengeceknya.

Bagaimana caranya?

Cukup dengan mengimpor file auth.php:

<?php require\_once("auth.php"); ?>

Apa bila user-nya belum login, maka dia akan dialihkan ke halaman login. Sesuai kondisi logika yang kita buat pada file auth.php.

Karena kita sudah mengimpor file auth.php, kita bisa menggunakan variabel $\_SESSION di sini.

Mari kita coba ambil nilai yang disimpan di sana:

<img class="img img-responsive rounded-circle mb-3" width="160" src="img/<?php echo $\_SESSION['user']['photo'] ?>" />

<h3><?php echo *$\_SESSION*["user"]["name"] ?></h3>

<p><?php echo $\_SESSION["user"]["email"] ?></p>

Lalu membuat sebuah status *dummy* “lorem ipsum” sebanyak 6:

<?php for(*$i*=0; *$i* < 6; *$i*++){ ?>

<div class="card mb-3">

<div class="card-body">

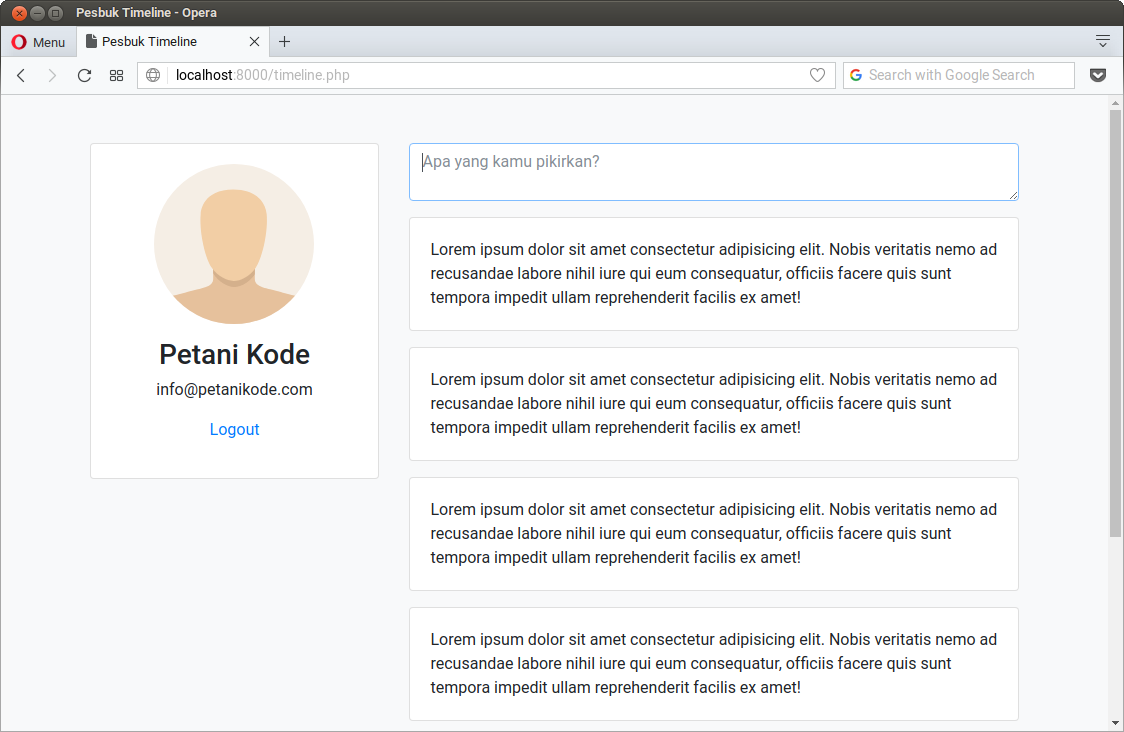
Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis veritatis nemo recusandae labore nihil iure qui eum consequatur, officiis facere quis sunt tempoimpedit ullam reprehenderit facilis ex amet!

</div>

</div>

<?php } ?>

Maka hasilnya:



Kereenn… 😍

Terakhir, tinggal buat logout.php saja nih.

**Membuat Logout di PHP**

Silahkan buka file logout.php, kemudian isi dengan kode berikut:

<?php

session\_start();

session\_unset("user");

header("Location: index.php");

Tungas dari file logout.php adalah menghancurkan atau menghapus *session* yang sudah dibuat.

Pertama-tama, kita harus memanggil fungsi session\_start(), karena kita akan menggunakan *session* di sini.

session\_start();

Setelah itu, kita hancurkan atau hapus variabel $\_SESSION['user'] dengan fungsi session\_unset("user").

session\_unset("user");

Penghancuran *session* selesai.

Selanjutnya kita alihkan ke halaman utama.

header("Location: index.php");