

Nama : Armanita Aning

NIM : 20.01.013.002

## 1. Cara Membuat CRUD

Create (C) merupakan proses pembuatan data baru. Proses ini biasanya dilakukan ketika Anda mendaftar pada sebuah halaman website. Data yang Anda masukkan akan disimpan di dalam database menggunakan operasi Create.

Read (R) merupakan proses pengambilan data dari database. Proses ini biasanya terjadi ketika Anda ingin melakukan proses login di halaman website tertentu. Saat Anda klik “Login” maka website akan menjalankan proses “Read” untuk melakukan verifikasi akun yang Anda gunakan.

Contoh operasi yang lebih sederhana yaitu ketika Anda mengakses sebuah halaman website, saat itu pula proses Read berjalan. Hasil dari operasi ini adalah tampilan halaman website yang Anda lihat di browser.

Update (U) adalah proses mengubah data yang berada di dalam database. Contoh proses ini ketika Anda mengubah profil di dalam akun sosial media. Saat Anda klik “Ubah”, website akan mengirimkan proses Update ke dalam database. Kemudian database meresponnya dengan mengubah data lama menjadi data baru yang Anda tambahkan melalui halaman profil.

Delete (D) adalah proses untuk menghapus data yang ada di database. Proses ini mirip dengan Update, bedanya, proses Delete akan mengubah data yang ada di database menjadi ‘tidak ada’.

Itulah gambaran kecil penggunaan CRUD di website.

## Coding Membuat Halaman Create

```
<?php

// Include config file
require_once "config.php";

// Define variables and initialize with empty values

$name = $address = $salary = "";
$name_err = $address_err = $salary_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Validate name

    $input_name = trim($_POST["name"]);

    if(empty($input_name)){

        $name_err = "Please enter a name.";

    } elseif(!filter_var($input_name, FILTER_VALIDATE_REGEXP,
        array("options">=>array("regexp">="/^[a-zA-Z\s]+$/"))) {

        $name_err = "Please enter a valid name.";

    } else{

        $name = $input_name;

    }

    // Validate address

    $input_address = trim($_POST["address"]);
```

```

if(empty($input_address)){
    $address_err = "Please enter an address.";
} else{
    $address = $input_address;
}

// Validate salary
$input_salary = trim($_POST["salary"]);
if(empty($input_salary)){
    $salary_err = "Please enter the salary amount.";
} elseif(!ctype_digit($input_salary)){
    $salary_err = "Please enter a positive integer value.";
} else{
    $salary = $input_salary;
}

// Check input errors before inserting in database
if(empty($name_err) && empty($address_err) && empty($salary_err)){
    // Prepare an insert statement
    $sql = "INSERT INTO employees (name, address, salary) VALUES (?, ?, ?)";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "sss", $param_name, $param_address, $param_salary);
    }
}

```

```
// Set parameters

$params_name = $name;

$params_address = $address;

$params_salary = $salary;


// Attempt to execute the prepared statement

if(mysqli_stmt_execute($stmt)){

    // Records created successfully. Redirect to landing page

    header("location: index.php");

    exit();

} else{

    echo "Something went wrong. Please try again later.";

}

}


// Close statement

mysqli_stmt_close($stmt);

}


// Close connection

mysqli_close($link);

}

?>


<!DOCTYPE html>
```

```
<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Create Record</title>

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

  <style type="text/css">

    .wrapper{

      width: 500px;

      margin: 0 auto;

    }

  </style>

</head>

<body>

  <div class="wrapper">

    <div class="container-fluid">

      <div class="row">

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

          <div class="page-header">

            <h2>Tambah Record</h2>

          </div>

          <p> Silahkan isi form di bawah ini kemudian submit untuk menambahkan data
pegawai ke dalam database.</p>

          <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">

            <div class="form-group <?php echo (!empty($name_err)) ? 'has-error' : ''; ?>">
```

```
<label>Nama</label>

<input type="text" name="name" class="form-control" value="<?php echo
$name; ?>">

<span class="help-block"><?php echo $name_err;?></span>

</div>

<div class="form-group <?php echo (!empty($address_err)) ? 'has-error' : "; ?>">

<label>Alamat</label>

<textarea name="address" class="form-control"><?php echo $address;
?></textarea>

<span class="help-block"><?php echo $address_err;?></span>

</div>

<div class="form-group <?php echo (!empty($salary_err)) ? 'has-error' : "; ?>">

<label>Salary</label>

<input type="text" name="salary" class="form-control" value="<?php echo
$salary; ?>">

<span class="help-block"><?php echo $salary_err;?></span>

</div>

<input type="submit" class="btn btn-primary" value="Submit">

<a href="index.php" class="btn btn-default">Cancel</a>

</form>

</div>

</div>

</div>

</div>

</body>

</html>
```

Baris kode di atas akan menampilkan form yang berfungsi untuk menambahkan data pegawai ke dalam database. Di dalam baris kode di atas juga terdapat validasi input sebelum data disimpan di dalam database.

### Coding Membuat Halaman Read

File yang bertugas untuk memasukkan data sudah ada. Tugas selanjutnya adalah membuat file yang bertugas untuk mengambil/membaca data dari database dan menampilkannya dalam bentuk tabel.

Silakan buat file dengan nama “read.php” kemudian masukkan baris kode di bawah ini ke dalamnya.

Baris kode di bawah akan menampilkan informasi secara detail masing-masing pegawai jika Anda mengklik icon read yang ada di dalam tabel.

```
<?php

// Check existence of id parameter before processing further
if(isset($_GET["id"]) && !empty(trim($_GET["id"]))) {

    // Include config file
    require_once "config.php";

    // Prepare a select statement
    $sql = "SELECT * FROM employees WHERE id = ?";
```

```

if($stmt = mysqli_prepare($link, $sql)){
    // Bind variables to the prepared statement as parameters
    mysqli_stmt_bind_param($stmt, "i", $param_id);

    // Set parameters
    $param_id = trim($_GET["id"]);

    // Attempt to execute the prepared statement
    if(mysqli_stmt_execute($stmt)){
        $result = mysqli_stmt_get_result($stmt);

        if(mysqli_num_rows($result) == 1){
            /* Fetch result row as an associative array. Since the result set
            contains only one row, we don't need to use while loop */
            $row =
            mysqli_fetch_array($result, MYSQLI_ASSOC);

            // Retrieve individual field value
            $name = $row["name"];
            $address = $row["address"];
            $salary = $row["salary"];
        } else{
            // URL doesn't contain valid id parameter. Redirect to error page
            header("location: error.php");
            exit();
        }
    }
}

```



```
    } else{  
        echo "Oops! Something went wrong. Please try again later.";  
    }  
}  
  
// Close statement  
mysqli_stmt_close($stmt);  
  
// Close connection  
mysqli_close($link);  
} else{  
    // URL doesn't contain id parameter. Redirect to error page  
    header("location: error.php");  
    exit();  
}  
?>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>View Record</title>  
    <link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">  
    <style type="text/css">
```

```
.wrapper{
    width: 500px;
    margin: 0 auto;
}
</style>
</head>
<body>
<div class="wrapper">
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-12">
                <div class="page-header">
                    <h1>View Record</h1>
                </div>
                <div class="form-group">
                    <label>Name</label>
                    <p class="form-control-static"><?php echo $row["name"]; ?></p>
                </div>
                <div class="form-group">
                    <label>Address</label>
                    <p class="form-control-static"><?php echo $row["address"]; ?></p>
                </div>
                <div class="form-group">
                    <label>Salary</label>
                    <p class="form-control-static"><?php echo $row["salary"]; ?></p>
                </div>
            </div>
        </div>
    </div>
</body>
```

```
</div>

<p><a href="index.php" class="btn btn-primary">Back</a></p>

</div>

</div>

</div>

</div>

</body>

</html>
```

## Membuat Halaman Update

Setelah membuat Create dan Read, sekarang giliran Anda untuk membuat file Update untuk mengubah data yang sudah disimpan. Silakan buat file “update.php” dan tambahkan baris kode di bawah ini ke dalamnya. Baris kode ini akan mengarahkan Anda ke halaman lain yang berisi form untuk mengubah informasi yang sudah ada di dalam database.

```
<?php

// Include config file

require_once "config.php";


// Define variables and initialize with empty values

$name = $address = $salary = "";

$name_err = $address_err = $salary_err = "";


// Processing form data when form is submitted
```

```
if(isset($_POST["id"]) && !empty($_POST["id"])){

    // Get hidden input value

    $id = $_POST["id"];


    // Validate name

    $input_name = trim($_POST["name"]);

    if(empty($input_name)){

        $name_err = "Please enter a name.";

    } elseif(!filter_var($input_name, FILTER_VALIDATE_REGEXP,
array("options"=>array("regexp"=>"/^[a-zA-Z\s]+$/")))){

        $name_err = "Please enter a valid name.";

    } else{

        $name = $input_name;

    }


    // Validate address address

    $input_address = trim($_POST["address"]);

    if(empty($input_address)){

        $address_err = "Please enter an address.";

    } else{

        $address = $input_address;

    }


    // Validate salary

    $input_salary = trim($_POST["salary"]);
```

```

if(empty($input_salary)){

    $salary_err = "Please enter the salary amount.";

} elseif(!ctype_digit($input_salary)){

    $salary_err = "Please enter a positive integer value.";

} else{

    $salary = $input_salary;

}


// Check input errors before inserting in database

if(empty($name_err) && empty($address_err) && empty($salary_err)){

    // Prepare an update statement

    $sql = "UPDATE employees SET name=?, address=?, salary=? WHERE id=?";

    if($stmt = mysqli_prepare($link, $sql)){

        // Bind variables to the prepared statement as parameters

        mysqli_stmt_bind_param($stmt, "sssi", $param_name, $param_address, $param_salary,
$param_id);

        // Set parameters

        $param_name = $name;

        $param_address = $address;

        $param_salary = $salary;

        $param_id = $id;

        // Attempt to execute the prepared statement

```

```

        if(mysqli_stmt_execute($stmt)){

            // Records updated successfully. Redirect to landing page

            header("location: index.php");

            exit();

        } else{

            echo "Something went wrong. Please try again later.";

        }

    }

}

// Close statement

mysqli_stmt_close($stmt);

}

// Close connection

mysqli_close($link);

} else{

    // Check existence of id parameter before processing further

    if(isset($_GET["id"]) && !empty(trim($_GET["id"]))) {

        // Get URL parameter

        $id = trim($_GET["id"]);

        // Prepare a select statement

        $sql = "SELECT * FROM employees WHERE id = ?";

        if($stmt = mysqli_prepare($link, $sql)){

            // Bind variables to the prepared statement as parameters

```

```

mysqli_stmt_bind_param($stmt, "i", $param_id);

// Set parameters

$param_id = $id;

// Attempt to execute the prepared statement
if(mysqli_stmt_execute($stmt)){
    $result = mysqli_stmt_get_result($stmt);

    if(mysqli_num_rows($result) == 1){
        /* Fetch result row as an associative array. Since the result set
           contains only one row, we don't need to use while loop */
        $row =
        mysqli_fetch_array($result, MYSQLI_ASSOC);

        // Retrieve individual field value

        $name = $row["name"];

        $address = $row["address"];

        $salary = $row["salary"];
    } else{
        // URL doesn't contain valid id. Redirect to error page

        header("location: error.php");

        exit();
    }

} else{

```

```
        echo "Oops! Something went wrong. Please try again later.";
    }
}

// Close statement
mysqli_stmt_close($stmt);

// Close connection
mysqli_close($link);
} else{
    // URL doesn't contain id parameter. Redirect to error page
    header("location: error.php");
    exit();
}
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Update Record</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
    <style type="text/css">
```





```
<textarea name="address" class="form-control"><?php echo $address;
?></textarea>

<span class="help-block"><?php echo $address_err;?></span>

</div>

<div class="form-group <?php echo (!empty($salary_err)) ? 'has-error' : ''; ?>">

    <label>Salary</label>

    <input type="text" name="salary" class="form-control" value="<?php echo
$salary; ?>">

    <span class="help-block"><?php echo $salary_err;?></span>

</div>

<input type="hidden" name="id" value="<?php echo $id; ?>" />

<input type="submit" class="btn btn-primary" value="Submit">

<a href="index.php" class="btn btn-default">Cancel</a>

</form>

</div>

</div>

</div>

</div>

</body>

</html>
```

## Coding Membuat Halaman Delete

Supaya Anda dapat menghapus informasi/data pegawai yang sudah masuk ke dalam database, maka perlu sebuah fungsi untuk menghapusnya. Silahkan buat file dengan nama “delete.php” kemudian masukkan baris kode di bawah ini ke dalamnya. Baris kode ini akan menghapus data ketika Anda klik icon delete yang ada di dalam tabel.

```
<?php

// Process delete operation after confirmation
if(isset($_POST["id"]) && !empty($_POST["id"])){

    // Include config file
    require_once "config.php";

    // Prepare a delete statement
    $sql = "DELETE FROM employees WHERE id = ?";

    if($stmt = mysqli_prepare($link, $sql)){

        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "i", $param_id);

        // Set parameters
        $param_id = trim($_POST["id"]);

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){

            // Records deleted successfully. Redirect to landing page
            header("location: index.php");
```

```
        exit();

    } else{

        echo "Oops! Something went wrong. Please try again later.";

    }

}

// Close statement
mysqli_stmt_close($stmt);

// Close connection
mysqli_close($link);
} else{

    // Check existence of id parameter
    if(empty(trim($_GET["id"]))) {

        // URL doesn't contain id parameter. Redirect to error page
        header("location: error.php");

        exit();

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>View Record</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

    .wrapper{

        width: 500px;

        margin: 0 auto;

    }

</style>

</head>

<body>

    <div class="wrapper">

        <div class="container-fluid">

            <div class="row">

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

                    <div class="page-header">

                        <h1>Delete Record</h1>

                    </div>

                    <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">

                        <div class="alert alert-danger fade in">

                            <input type="hidden" name="id" value="<?php echo trim($_GET["id"]); ?>"/>

                            <p>Are you sure you want to delete this record?</p><br>

                            <p>

                                <input type="submit" value="Yes" class="btn btn-danger">

                                <a href="index.php" class="btn btn-default">No</a>

                            </p>

                        </div>

                    </form>

                </div>

            </div>

        </div>

    </div>

</body>

</html>
```

</div>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

## Membuat Halaman Error

Terkadang ada baris kode yang bermasalah. Supaya Anda tidak perlu mendeklarasikan halaman error pada setiap file, akan lebih baik jika Anda membuat halaman error sendiri. Halaman ini akan menampilkan pesan error jika terjadi logika atau baris kode yang bermasalah. Anda pun dapat menyesuaikan informasi di dalam error ini sesuai dengan kebutuhan.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Error</title>
```

```
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
```

```
  <style type="text/css">
```

```
    .wrapper{
```

```
      width: 750px;
```

```
      margin: 0 auto;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div class="wrapper">
```

```
    <div class="container-fluid">
```

```
      <div class="row">
```

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

```
          <div class="page-header">
```

```
            <h1>Invalid Request</h1>
```

</div>

<div class="alert alert-danger fade in">

<p>Sorry, you've made an invalid request. Please <a href="index.php" class="alert-link">go back</a> and try again.</p>

</div>

</div>

</div>

</div>

</div>

</body>

</html>



## Membuat CRUD di Halaman CRUD

Contoh di atas merupakan contoh sederhana penggunaan CRUD untuk menambahkan, mengubah, mengambil, dan menghapus data dari dan ke database. Penggunaan CRUD masih sangat luas. Salah satu yang sering dijumpai adalah ketika Anda melakukan login ke halaman website tertentu. Itu sebenarnya adalah contoh kecil dari penggunaan CRUD. Pada bagian ini akan membahas mengenai cara membuat halaman login menggunakan sistem autentifikasi dengan operasi CRUD. Terdengar sederhana, tapi cukup penting karena berhubungan dengan keamanan sebuah aplikasi web. Kenapa? Autentifikasi merupakan mekanisme yang digunakan untuk membatasi akses dari user yang tidak mempunyai hak akses atau tools/fitur tertentu. Sama seperti langkah sebelumnya, Anda juga harus mempersiapkan terlebih dahulu database yang berisi daftar user untuk menjalankan proses login. Artikel ini juga akan menggunakan PHP dan MySQL sebagai bahasa pemrograman utama dan database server-nya.

Pada bagian pertama Anda akan ditunjukkan cara membuat form user untuk proses registrasi. Kemudian membuat form login pada bagian kedua dan juga seperti biasanya, yaitu membuat halaman utama dan script untuk melakukan logout.

### 1. Membuat Sistem Registrasi

Bagian ini nanti Anda akan membuat sebuah form registrasi lengkap dengan database di dalamnya. User akan dapat melakukan registrasi hanya dengan mengisi form yang sudah tersedia. Jadi langkah pertama yang perlu dilakukan adalah menyiapkan database sebagai penyimpanan data user.

## Membuat Tabel Database

Jalankan perintah di bawah ini untuk menambahkan record ke dalam database yang sudah dibuat sebelumnya (di artikel ini contohnya database niagahoster). Anda bisa menyesuaikan jumlah variabel sesuai dengan kebutuhan. Namun ketika membuat tabel untuk user, hal yang paling penting adalah menambahkan variabel id, username, dan password. Ketiga variabel tersebut nantinya menjadi inti di dalam membuat form login menggunakan autentifikasi.

```
CREATE TABLE users (
```

```
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
```

```
    username VARCHAR(50) NOT NULL UNIQUE,
```

```
password VARCHAR(255) NOT NULL,  
created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

Pastikan bahwa tabel users sudah terbuat dengan baik.

## 2. Membuat File Config

Setelah membuat tabel, supaya bisa menjalankan proses CRUD, langkah selanjutnya adalah membuat jembatan penghubung antara PHP dengan MySQL. Anda hanya tinggal membuat script PHP yang bertugas untuk menghubungkan ke database server. Silahkan membuat file “config.php” (atau bisa menyalin dari program sebelumnya), kemudian masukkan baris kode di bawah ini.

```
<?php  
  
/* Database credentials. Assuming you are running MySQL  
server with default setting (user 'root' with no password) */define('DB_SERVER', 'localhost');  
define('DB_USERNAME', 'root');  
define('DB_PASSWORD', 'password');  
define('DB_NAME', 'niagahoster');  
  
/* Attempt to connect to MySQL database */$link = mysqli_connect(DB_SERVER,  
DB_USERNAME, DB_PASSWORD, DB_NAME);  
  
// Check connection  
if($link === false){
```

```

        die("ERROR: Could not connect. " . mysqli_connect_error());
    }
?>

```

Ganti informasi di atas sesuai dengan pengaturan di MySQL server Anda. Ganti username, password, dan nama database seperti yang Anda buat.

### 3. Membuat Form Registrasi

Setelah data sudah siap dan PHP dapat terhubung dengan MySQL, langkah selanjutnya adalah membuat tampilan front-end untuk menampilkan hasil operasinya. Buat file dengan nama “register.php” kemudian tambahkan baris kode di bawah ini. Namun perlu Anda ketahui bahwa baris kode di bawah ini mengizinkan user untuk membuat akunnya sendiri (tanpa perlu proses verifikasi).

```

<?php

// Include config file
require_once "config.php";

// Define variables and initialize with empty values

$username = $password = $confirm_password = "";
$username_err = $password_err = $confirm_password_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Validate username

    if(empty(trim($_POST["username"]))) {

        $username_err = "Please enter a username.";
    }
}

```

```

} else{

    // Prepare a select statement

    $sql = "SELECT id FROM users WHERE username = ?";

    if($stmt = mysqli_prepare($link, $sql)){

        // Bind variables to the prepared statement as parameters

        mysqli_stmt_bind_param($stmt, "s", $param_username);

        // Set parameters

        $param_username = trim($_POST["username"]);

        // Attempt to execute the prepared statement

        if(mysqli_stmt_execute($stmt)){

            /* store result */            mysqli_stmt_store_result($stmt);

            if(mysqli_stmt_num_rows($stmt) == 1){

                $username_err = "This username is already taken.";

            } else{

                $username = trim($_POST["username"]);

            }

        } else{

            echo "Oops! Something went wrong. Please try again later.";

        }

    }

}

```

```
// Close statement

mysqli_stmt_close($stmt);

}


// Validate password

if(empty(trim($_POST["password"]))) {

    $password_err = "Please enter a password.";

} elseif(strlen(trim($_POST["password"])) < 6) {

    $password_err = "Password must have atleast 6 characters.";

} else {

    $password = trim($_POST["password"]);

}


// Validate confirm password

if(empty(trim($_POST["confirm_password"]))) {

    $confirm_password_err = "Please confirm password.";

} else {

    $confirm_password = trim($_POST["confirm_password"]);

    if(empty($password_err) && ($password != $confirm_password)) {

        $confirm_password_err = "Password did not match.";

    }

}


// Check input errors before inserting in database

if(empty($username_err) && empty($password_err) && empty($confirm_password_err)) {
```

```

// Prepare an insert statement

$sql = "INSERT INTO users (username, password) VALUES (?, ?)";

if($stmt = mysqli_prepare($link, $sql)){
    // Bind variables to the prepared statement as parameters

    mysqli_stmt_bind_param($stmt, "ss", $param_username, $param_password);

    // Set parameters

    $param_username = $username;

    $param_password = password_hash($password, PASSWORD_DEFAULT); // Creates a
password hash


    // Attempt to execute the prepared statement
    if(mysqli_stmt_execute($stmt)){
        // Redirect to login page

        header("location: login.php");
    } else{
        echo "Something went wrong. Please try again later.";
    }
}

// Close statement

mysqli_stmt_close($stmt);
}

```

```

// Close connection

mysqli_close($link);
}
?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Sign Up</title>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

    <style type="text/css">

        body{ font: 14px sans-serif; }

        .wrapper{ width: 350px; padding: 20px; }

    </style>

</head>

<body>

    <div class="wrapper">

        <h2>Sign Up</h2>

        <p>Please fill this form to create an account.</p>

        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">

            <div class="form-group <?php echo (!empty($username_err)) ? 'has-error' : ''; ?>">

                <label>Username</label>

```

```
<input type="text" name="username" class="form-control" value="<?php echo
$username; ?>">

<span class="help-block"><?php echo $username_err; ?></span>

</div>

<div class="form-group <?php echo (!empty($password_err)) ? 'has-error' : ''; ?>">

    <label>Password</label>

    <input type="password" name="password" class="form-control" value="<?php echo
$password; ?>">

    <span class="help-block"><?php echo $password_err; ?></span>

</div>

<div class="form-group <?php echo (!empty($confirm_password_err)) ? 'has-error' : '';
?>">

    <label>Confirm Password</label>

    <input type="password" name="confirm_password" class="form-control"
value="<?php echo $confirm_password; ?>">

    <span class="help-block"><?php echo $confirm_password_err; ?></span>

</div>

<div class="form-group">

    <input type="submit" class="btn btn-primary" value="Submit">

    <input type="reset" class="btn btn-default" value="Reset">

</div>

<p>Already have an account? <a href="login.php">Login here</a>.</p>

</form>

</div>

</body>

</html>
```



Contoh di atas menggunakan fungsi PHP `'password_hash()'` yang berfungsi untuk mengubah dari password string menjadi teks hash.

Hal ini cukup menarik karena fungsi tersebut menggenerate sebuah password menggunakan one-way hashing algoritma yang cukup kuat. Fungsi ini juga melakukan sistem random salt automatically (pengacakan garam) ketika melakukan hashing password.

Proses hashing yang dilakukan tersebut memungkinkan penggunaan satu password yang sama. Namun karena ada proses hashing ini, password bisa sama, tapi kode hash akan tetap berbeda.

Selain itu, baris kode di atas akan menampilkan form untuk registrasi sekaligus pengecekan error. Pengguna yang tidak mengisi data kosong atau tidak lengkap akan menerima notifikasi error dan data tidak disimpan. Error ini juga berguna saat user ingin membuat username yang sudah ada.

Tampilan baris kode di atas kurang lebih seperti di bawah ini.

#### 4. Membuat Sistem Login

Setelah Anda memasukkan data dan membuat file konfigurasi, langkah selanjutnya adalah membuat bagian login dan halaman utama. Nantinya di dalam form login Anda dapat memasukkan username dan password sesuai dengan yang sudah dibuat di form registrasi.

Ketika user melakukan submit (login) maka sistem akan melalui verifikasi ke dalam data yang sudah disimpan di dalam database. Jika username dan password sesuai, user dapat dipercaya dan bisa langsung masuk ke dalam sistem. Namun jika salah maka user akan mendapatkan informasi penolakan (rejected).

Berikut ini langkah-langkah untuk membuat form login dan atribut lainnya.

## 5. Membuat Form Login

Pertama, buat terlebih dahulu file “login.php” dan masukkan kode di bawah ini ke dalamnya.

```
<?php

// Initialize the session

session_start();


// Check if the user is already logged in, if yes then redirect him to welcome page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
    header("location: welcome.php");
    exit;
}


// Include config file
require_once "config.php";


// Define variables and initialize with empty values
$username = $password = "";
$username_err = $password_err = "";


// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Check if username is empty
```

```

if(empty(trim($_POST["username"]))) {
    $username_err = "Please enter username.";
} else {
    $username = trim($_POST["username"]);
}

// Check if password is empty
if(empty(trim($_POST["password"]))) {
    $password_err = "Please enter your password.";
} else {
    $password = trim($_POST["password"]);
}

// Validate credentials
if(empty($username_err) && empty($password_err)) {
    // Prepare a select statement
    $sql = "SELECT id, username, password FROM users WHERE username = ?";

    if($stmt = mysqli_prepare($link, $sql)) {
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $param_username);

        // Set parameters
        $param_username = $username;
    }
}

```

```

// Attempt to execute the prepared statement
if(mysqli_stmt_execute($stmt)){

    // Store result

    mysqli_stmt_store_result($stmt);


    // Check if username exists, if yes then verify password
    if(mysqli_stmt_num_rows($stmt) == 1){

        // Bind result variables

        mysqli_stmt_bind_result($stmt, $id, $username, $hashed_password);

        if(mysqli_stmt_fetch($stmt)){

            if(password_verify($password, $hashed_password)){

                // Password is correct, so start a new session

                session_start();


                // Store data in session variables

                $_SESSION["loggedin"] = true;

                $_SESSION["id"] = $id;

                $_SESSION["username"] = $username;


                // Redirect user to welcome page

                header("location: welcome.php");

            } else{

                // Display an error message if password is not valid

                $password_err = "The password you entered was not valid.";

            }
        }
    }
}

```

```
    }

    } else{

        // Display an error message if username doesn't exist

        $username_err = "No account found with that username.";

    }

    } else{

        echo "Oops! Something went wrong. Please try again later.";

    }

}

// Close statement

mysqli_stmt_close($stmt);

}

// Close connection

mysqli_close($link);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Login</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

<style type="text/css">

    body{ font: 14px sans-serif; }

    .wrapper{ width: 350px; padding: 20px; }

</style>

</head>

<body>

    <div class="wrapper">

        <h2>Login</h2>

        <p>Please fill in your credentials to login.</p>

        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">

            <div class="form-group <?php echo (!empty($username_err)) ? 'has-error' : ''; ?>">

                <label>Username</label>

                <input type="text" name="username" class="form-control" value="<?php echo
$username; ?>">

                <span class="help-block"><?php echo $username_err; ?></span>

            </div>

            <div class="form-group <?php echo (!empty($password_err)) ? 'has-error' : ''; ?>">

                <label>Password</label>

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

                <span class="help-block"><?php echo $password_err; ?></span>

            </div>

            <div class="form-group">

                <input type="submit" class="btn btn-primary" value="Login">

            </div>

        </div>

    </body>

</html>
```

```
</div>

<p>Don't have an account? <a href="register.php">Sign up now</a>.</p>

</form>

</div>

</body>

</html>
```

Baris kode di atas akan mengambil input id, username, dan password yang dimasukkan user kemudian memverifikasi ke dalam tabel 'users' di dalam database.

Jika password dan username yang dimasukkan sesuai, user akan langsung masuk dan menampilkan halaman (session) baru –pada kasus ini kita memakai file "welcome.php".

Baris kode di atas kurang lebih akan menampilkan tampilan seperti ini.

## 6. Membuat Halaman Utama

Setelah form login selesai tentu saja membuat halaman utama untuk ditampilkan saat user berhasil masuk ke dalam sistem.

Nah! Terkadang pada proses login terdapat data yang berasal dari sumber yang tidak dipercaya yang mencoba untuk masuk ke dalam sistem. Biasanya ini diisi oleh user anonymous. Ini sangat beresiko karena terkadang mengandung script yang berbahaya contohnya serangan cross-site scripting (XSS).

Supaya Anda terhindar dari XSS, PHP menyediakan fungsi yang berguna untuk menangkulangnya yaitu menggunakan fungsi htmlspecialchars() sebelum menampilkan halaman di browser. Fungsi ini membuat segala jenis tag HTML yang mengandung script 'jahat' menjadi tidak berbahaya.

Untuk membuat file ini, silakan buat file "welcome.php" kemudian masukkan baris kode di bawah.

```
<?php

// Initialize the session

session_start();


// Check if the user is logged in, if not then redirect him to login page
if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){

    header("location: login.php");

    exit;

}

?>


<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Welcome</title>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">

    <style type="text/css">

        body{ font: 14px sans-serif; text-align: center; }

    </style>

</head>

<body>

    <div class="page-header">

        <h1>Hi, <b><?php echo htmlspecialchars($_SESSION["username"]); ?></b>. Welcome to
our site.</h1>
```



```
</div>

<p>

    <a href="reset-password.php" class="btn btn-warning">Reset Your Password</a>

    <a href="logout.php" class="btn btn-danger">Sign Out of Your Account</a>

</p>

</body>

</html>
```

## 7. Membuat Script Logout

Supaya user dapat berpindah dari akun satu ke akun yang lainnya, Anda perlu membuat fitur log out. Fungsi dari file ini adalah menutup halaman (session) ketika user ini keluar dari akunnya. Di dalam script ini terdapat sebuah fungsi yang berjalan ketika user melakukan klik pada tombol atau link log out atau sign out. Script akan menutup session dan kemudian melakukan redirect ke halaman login.

Silakan buat file dengan nama “logout.php” kemudian masukkan baris kode di bawah ini.

```
<?php

// Initialize the session

session_start();


// Unset all of the session variables

$_SESSION = array();


// Destroy the session.

session_destroy();
```

```
// Redirect to login page  
header("location: login.php");  
exit;  
?>
```

## 8. Membuat Fitur Reset Password

Opsional! Anda bisa melanjutkan ke bagian ini jika ingin menambahkan fitur reset password di dalam aplikasi.

```
<?php  
  
// Initialize the session  
session_start();  
  
// Check if the user is logged in, otherwise redirect to login page  
if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){  
    header("location: login.php");  
    exit;  
}  
  
// Include config file  
require_once "config.php";  
  
// Define variables and initialize with empty values  
$new_password = $confirm_password = "";
```

```
$new_password_err = $confirm_password_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Validate new password
    if(empty(trim($_POST["new_password"]))){
        $new_password_err = "Please enter the new password.";
    } elseif(strlen(trim($_POST["new_password"])) < 6){
        $new_password_err = "Password must have atleast 6 characters.";
    } else{
        $new_password = trim($_POST["new_password"]);
    }

    // Validate confirm password
    if(empty(trim($_POST["confirm_password"]))){
        $confirm_password_err = "Please confirm the password.";
    } else{
        $confirm_password = trim($_POST["confirm_password"]);
        if(empty($new_password_err) && ($new_password != $confirm_password)){
            $confirm_password_err = "Password did not match.";
        }
    }
}

// Check input errors before updating the database
```

```

if(empty($new_password_err) && empty($confirm_password_err)){

    // Prepare an update statement

    $sql = "UPDATE users SET password = ? WHERE id = ?";

    if($stmt = mysqli_prepare($link, $sql)){

        // Bind variables to the prepared statement as parameters

        mysqli_stmt_bind_param($stmt, "si", $param_password, $param_id);

        // Set parameters

        $param_password = password_hash($new_password, PASSWORD_DEFAULT);

        $param_id = $_SESSION["id"];

        // Attempt to execute the prepared statement

        if(mysqli_stmt_execute($stmt)){

            // Password updated successfully. Destroy the session, and redirect to login page

            session_destroy();

            header("location: login.php");

            exit();

        } else{

            echo "Oops! Something went wrong. Please try again later.";

        }

    }

    // Close statement

    mysqli_stmt_close($stmt);

```

```
}
```

```
// Close connection
```

```
mysqli_close($link);
```

```
}
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Reset Password</title>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
```

```
<style type="text/css">
```

```
body{ font: 14px sans-serif; }
```

```
.wrapper{ width: 350px; padding: 20px; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="wrapper">
```

```
<h2>Reset Password</h2>
```

```
<p>Please fill out this form to reset your password.</p>
```

```
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">
```

```
<div class="form-group <?php echo (!empty($new_password_err)) ? 'has-error' : ''; ?>">
```

```
<label>New Password</label>

<input type="password" name="new_password" class="form-control" value="<?php
echo $new_password; ?>">

<span class="help-block"><?php echo $new_password_err; ?></span>

</div>

<div class="form-group <?php echo (!empty($confirm_password_err)) ? 'has-error' : ";
?>">

<label>Confirm Password</label>

<input type="password" name="confirm_password" class="form-control">

<span class="help-block"><?php echo $confirm_password_err; ?></span>

</div>

<div class="form-group">

<input type="submit" class="btn btn-primary" value="Submit">

<a class="btn btn-link" href="welcome.php">Cancel</a>

</div>

</form>

</div>

</body>

</html>
```