

# Jobsheet-07 PHP – Form Processing

Charellino Kalingga Sadewo / 07 / 2341720205

Github: <https://github.com/CrushedKatana/DesProWeb>

No1:

```
<?php
$umur;
if (isset($umur)&& $umur >= 18){
    echo "Anda sudah dewasa.";
}else{
    echo "Anda belum dewasa / variable 'umur' tidak ditemukan. ";
}
?>
```

Anda belum dewasa / variable 'umur' tidak ditemukan.

The PHP code will display the result "You are not an adult or the variable 'age' is not found". This happens because the variable \$age is only declared without being given a value, so the condition `isset($age)` will be false. The `isset()` function checks whether the variable has been initialized and is not null. Because \$age has not been initialized with a value, the else section will be executed.

No2:

```
<?php
$umur;
if (isset($umur)&& $umur >= 18){
    echo "Anda sudah dewasa.";
}else{
    echo "Anda belum dewasa / variable 'umur' tidak ditemukan. ";
}

echo"<br>";

$data = array ("nama" => "Jane", "usia" =>25 );
if (isset($data["nama"])){
    echo "Nama: " . $data ["nama"];
} else {
    echo "Variable 'nama' tidak ditemukan dalam array.";
}
```

```
?>
```

Anda belum dewasa / variable 'umur' tidak ditemukan.

Nama: Jane

The PHP code will display "Name: Jane". This is because the \$data array contains key-value pairs, where the key "name" has the value "Jane". The `isset($data["name"])` function checks whether the element "name" exists in the array, and returns true. Since the element exists, the if block will execute, displaying "Name: Jane".

No3:

```
<?php
$myArray = array();
if (empty($myArray)) {
    echo "Array tidak terdefinisi / kosong.";
} else {
    echo "Array terdefinisi dan tidak kosong.";
}
?>
```

Array tidak terdefinisi / kosong.

The PHP code will display "Array is undefined or empty". This is because the `empty($myArray)` function checks whether the array \$myArray is empty or undefined. In this case, \$myArray is an empty array, so `empty()` returns true. Therefore, the if block is executed and the message "Array is undefined or empty" is displayed.

No4:

```
<?php
$myArray = array();
if (empty($myArray)) {
    echo "Array tidak terdefinisi / kosong.";
} else {
    echo "Array terdefinisi dan tidak kosong.";
}
echo "<br>";

if (empty($nonExstedVar)) {
    echo "Variabel tidak terdefinisi dan kosong.";
} else {
    echo "Variabel tidak terdefinisi dan kosong.";
}
```

```
?>
```

Array tidak terdefinisi / kosong.  
Variabel tidak terdefinisi dan kosong.

The PHP code will display "Variable is undefined or empty". This is because the variable `$nonExistentVar` has never been declared, so it is considered undefined. The `empty()` function checks whether the variable does not exist or has an empty value (such as `null`, `0`, `""`, etc.), and if so, it returns `true`. Since the variable is undefined, the `if` block is executed and the message is displayed.

No5:

```
<!DOCTYPE html>
<html>
<head>
    <title>Form Input PHP</title>
</head>
<body>
    <h2>Form Input PHP</h2>
    <form>method ="post " action ="proses_form.php"
    <label for="nama">Nama: </label>
    <input type="text" nama="nama" id="nama" required> <br><br>

    <label for="email">Email:</label>
    <input type="email" name="email" id="email" required> <br><br>

    <input type="submit" name="submit" value="Submit">
    </form>
</body>
</html>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];

    echo "Nama: " . $nama . "<br>";
    echo "Email: " . $email;
}
?>
```

## Form Input PHP

Nama:

Email:

Nama: Charellino Kalingga Sadewo

Email: charellinokalinggasadewo@gmail.com

The PHP and HTML code creates a simple input form with the POST method. When the user fills in the form with their name and email, then sends it by pressing the Submit button, the data will be sent to the proses\_form.php file. In the proses\_form.php file, PHP will process the input using \$\_POST, and then display the name and email entered. If the form is filled in with "Charellino Kalingga Sadewo" as the name and "example@gmail.com" as the email, the result will display "Name: Charellino Kalingga Sadewo" and "Email: charellinokalinggasadewo@gmail.com" on the page.

No6:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Form Input PHP</title>
</head>

<body>
    <h2>Form Input PHP</h2>
    <?php
        // insialisasi varriabel
        $namaErr = "";
        $nama = "";

        //cek apakah form sudah disubmit
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            if (empty($_POST["nama"])) {
                $namaErr = "Nama harus diisi!";
            } else {
                $nama = $_POST["nama"];
                echo "Data berhasil disimpan!";
            }
        }
    ?>

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

```

<label for="nama">Nama: </label>
<input type="text" name="nama" id="nama" value="<?php echo $nama;
?>">

<span class="error">
    <?php echo $namaErr; ?>
</span><br><br>

<input type="submit" name="submit" id="submit">
</form>
</body>

</html>

```

### Form Input PHP

Nama:

### Form Input PHP

Data berhasil disimpan!

Nama:

### Form Input PHP

Nama:  Nama harus diisi!

The HTML and PHP code creates a simple input form with validation for the name field. If the user submits the form without filling in the name field, the error message "Name is required!" will appear next to the input field. If the name field is filled in and the form is submitted, then message "Data saved successfully!" will be displayed, and the entered name will remain visible in the input field after the form is submitted. This is done by using `value="<?php echo $name; ?>"` to keep the data in the input field after submission.

No7:

```

<?php
$input = $_POST['input'];
$input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Input</title>
</head>
<body>

```

```

<form action="html_safe.php" method="post">
    <label for="input">Enter some text (including HTML
tags):</label><br>
    <input type="text" id="input" name="input"><br><br>
    <input type="submit" value="Submit">
</form>
</body>
</html>

```

Masukkan Nama :



The PHP and HTML code will display a form where the user can enter text. Once the form is submitted (by pressing the Submit button), the user input is captured and secured using the htmlspecialchars() function to prevent special characters such as < and > from being interpreted as HTML. The sanitized input is then displayed back on the screen. If the user were to enter text such as "<script>alert('XSS')</script>", the text would be displayed as is without executing the JavaScript code, because the special characters have been made safe.

No8:

```

<?php
$input = $_POST['input'];
$input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
$email = $_POST['email'];

if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "Processed Input: " . $input . "<br>";
    echo "Valid Email: " . htmlspecialchars($email, ENT_QUOTES, 'UTF-8');
} else {
    echo "Processed Input: " . $input . "<br>";
    echo "Invalid Email!";
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Form Input</title>

```

```

</head>
<body>
    <form action="html_safe.php" method="post">
        <label for="input">Enter some text (including HTML
tags) :</label><br>
        <input type="text" id="input" name="input"><br><br>

        <label for="email">Enter your email:</label><br>
        <input type="email" id="email" name="email"><br><br>

        <input type="submit" value="Submit">
    </form>
</body>
</html>

```

Enter your name:

Enter your email:



Processed Input: Charellino Kalingga S  
Valid Email: charellinokalinggasadewo@gmail.com

Observation:

Valid email:

If you enter a valid email the output will display the escaped version of the text input and the valid email.

Invalid email:

If you enter an invalid email, the output will display the escaped version of the text input and the message "Invalid Email!".

Explanation:

The htmlspecialchars function continues to protect the text input from being treated as executable code, ensuring it is displayed safely as plain text.

The filter\_var function ensures that only valid email formats are accepted. If the email is invalid, a user-friendly message is shown instead of continuing with the processing.

No9:

```

<?php
$pattern = '/[a-z]/'; // cocokkan huruf kecil
$text = 'This is a Sample Text.';
if (preg_match($pattern, $text)) {

```

```

        echo "Huruf kecil ditemukan!";
        echo "<br>";
    } else {
        echo "tidak ada huruf kecil!";
        echo "<br>";
    }
?>

```

Huruf kecil ditemukan!

The PHP code uses the `preg_match()` function to search for lowercase letters in the string `$text`, which contains "This is a Sample Text". The regex pattern used is `'/[a-z]/'`, which searches for lowercase characters from 'a' to 'z'. Since the string contains lowercase letters, such as 'h', 'i', and 's', the result will display "Lowercase found!". If there are no lowercase letters in the string, then the output will be "No lowercase letters!".

No10:

```

$pattern = '/[0-9]+/'; // cocokkan satu atau lebih digit
$text = 'There are 123 apples..';
if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
    echo "<br>";
} else {
    echo "Tidak ada yang cocok!";
    echo "<br>";
}
?>

```

Huruf kecil ditemukan!  
Cocokkan: 123

The PHP code uses the `preg_match()` function to find if there is a digit (number) in the string `$text`, which contains "There are 123 apples". The regex pattern used is `'/[0-9]/'`, which looks for a single digit number from '0' to '9'. Since the string contains a number (123), the result will display "Match: 1", because `preg_match()` only captures the first match, which is the number 1. If there are no numbers in the string, it will display "No matches!".

No11:

```

$pattern = '/apple/';
$replacement = 'banana';
$text = 'I like apple pie.';

```



```
$new_text = preg_replace($pattern, $replacement, $text);
echo $new_text;
echo "<br>";

?>
```

Huruf kecil ditemukan!  
 Cocokkan: 123  
 I like banana pie.

The PHP code uses the `preg_replace()` function to replace the word "apple" with "banana" in the string `$text`, which contains "I like apple pie". The regex pattern used is `/apple/`, which searches for the word "apple" in the text. The function then replaces the word "apple" with "banana", so that the new text produced is "I like banana pie". The final result will be displayed as "I like banana pie" on the screen.

No12:

```
$pattern = '/go*d/'; // cocokkan "god", "good", "goood", dll.
$text = 'god is good';
if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
    echo "<br>";
} else {
    echo "Tidak ada yang cocok!";
    echo "<br>";
}
```

Huruf kecil ditemukan!  
 Cocokkan: 123  
 I like banana pie.  
 Cocokkan: god

The PHP code uses the `preg_match()` function with the regex pattern `/go*d/` which will match the words "god", "good", "goood", and so on (where there are 0 or more "o" letters between "g" and "d"). The string being searched for is "god is good". This regex pattern will find the first match, which is "god", because it matches the pattern. As a result, the program will display "Match: god". If there is no match, then what is displayed is "No match", but in this case there is a match.

No13:

```
$pattern = '/go?d/'; // cocokkan "god", "good", "goood", dll.
```

```

$text = 'god is good';
if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
    echo "<br>";

} else {
    echo "Tidak ada yang cocok!";
    echo "<br>";
}

```

Huruf kecil ditemukan!

Cocokkan: 123

I like banana pie.

Cocokkan: god

Cocokkan: god

The code uses the regex pattern `/go?d/`, which will match the character 'g', followed by 0 or 1 'o's, and ending with a 'd'. In the string "god is good", this pattern will match the word "god", because "o" appears once (according to the ``?`` rule which means 0 or 1 times). Therefore, the result is "Match: god". This regex will not match "good" because "o" appears twice, and the ``?`` rule only allows 0 or 1 occurrences of "o".

No14:

```

$pattern = '/go{2,4}d/'; // cocokkan "god", "good", "goood", dll.
$text = 'god is good';
if (preg_match($pattern, $text, $matches)) {
    echo "Cocokkan: " . $matches[0];
    echo "<br>";

} else {
    echo "Tidak ada yang cocok!";
    echo "<br>";
}

```

Huruf kecil ditemukan!

Cocokkan: 123

I like banana pie.

Cocokkan: god

Cocokkan: god

Cocokkan: good

The code uses the regex pattern `/go{1,2}d/`, which matches the character 'g', followed by 1 or 2 'o's, and ending with a 'd'. In the string "god is good and goood", this regex will match both "god" and "good", but only pick the first match it finds, which is "god". Since the regex looks for 1 to 2 occurrences of the letter 'o', the word "goood" (with 3 'o's) will not be matched. The result is "Match: god".

No15:

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contoh Form dengan PHP</title>
</head>

<body>
  <h2>Form Contoh</h2>
  <form method="post" action="proses_lanjut.php">
    <label for="buah">Pilih Buah: </label>
    <select name="buah" id="buah">
      <option value="apel">Apel</option>
      <option value="pisang">Pisang</option>
      <option value="mangga">Mangga</option>
      <option value="jeruk">Jeruk</option>
    </select>

    <br>

    <label>Pilih Warna Favorit: </label><br>
    <input type="checkbox" name="warna[]" value="merah"> Merah <br>
    <input type="checkbox" name="warna[]" value="biru"> Biru <br>
    <input type="checkbox" name="warna[]" value="hijau"> Hijau <br>

    <br>

    <label>Pilih Jenis Kelamin: </label><br>
    <input type="radio" name="jenis_kelamin" value="laki-laki">
    Laki-laki <br>
```

```
        <input type="radio" name="jenis_kelamin" value="perempuan">
Perempuan <br>

        <br>
        <input type="submit" value="submit">

    </form>
</body>

</html>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $selectedBuah = $_POST['buah'];

    if (isset($_POST['warna'])) {
        $selectedWarna = $_POST['warna'];
    } else {
        $selectedWarna = [];
    }
    $selectedJenisKelamin = $_POST['jenis_kelamin'];
    echo "Anda memilih buah: " . $selectedBuah . "<br>";

    if (!empty($selectedWarna)) {
        echo "Warna favorit anda: " . implode(", ", $selectedWarna) .
"<br>";
    } else {
        echo "Anda tidak memilih warna favorit.<br>";
    }
    echo "Jenis kelamin Anda: " . $selectedJenisKelamin;
}
?>
```

## Form Contoh

Pilih Buah:

Pilih Warna Favorit:

- ☒ Merah
- ☒ Biru
- ☒ Hijau

Pilih Jenis Kelamin:

- ☒ Laki-laki
- ☐ Perempuan

Anda memilih buah: mangga  
Warna favorit anda: merah, biru, hijau  
Jenis kelamin Anda: laki-laki

The code generates an HTML form that allows the user to select a fruit, some favorite colors, and a gender. When the form is submitted, the data from the form is processed by a PHP script. The script will display the selected fruit, the selected favorite colors (if any), and the selected gender. If no colors are selected, the message "You did not select a favorite color" will appear.

No16:

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contoh Form dengan PHP dan jQuery</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>

<body>
  <h2>Form Contoh</h2>
  <form id="myForm">
    <label for="buah">Pilih Buah: </label>
    <select name="buah" id="buah">
      <option value="apel">Apel</option>
      <option value="pisang">Pisang</option>
      <option value="mangga">Mangga</option>
      <option value="jeruk">Jeruk</option>
    </select>

    <br>

    <label>Pilih Warna Favorit: </label><br>
```

```

<input type="checkbox" name="warna[]" value="merah"> Merah <br>
<input type="checkbox" name="warna[]" value="biru"> Biru <br>
<input type="checkbox" name="warna[]" value="hijau"> Hijau <br>

<br>

<label>Pilih Jenis Kelamin: </label><br>
<input type="radio" name="jenis_kelamin" value="laki-laki">
Laki-laki <br>
<input type="radio" name="jenis_kelamin" value="perempuan">
Perempuan <br>

<br>
<input type="submit" value="submit">

</form>

<div id="hasil">
  <!-- hasil akan ditampilkan di sini -->
</div>

<script>
$(document).ready(function () {
  $("#myForm").submit(function (e) {
    e.preventDefault(); //mencegah pengiriman form scr default

    //mengumpulkan data
    var formData = $("#myForm").serialize();

    //kirim data ke server PHP
    $.ajax({
      url: "proses_lanjut.php", // ganti nama dengan nama
      type: "POST",
      data: formData,
      success: function (response) {
        //tampil hasil dari server di div "hasil"
        $("#hasil").html(response);
      }
    });
  });
});

```

```

        });
    });
</script>
</body>

</html>

```

## Form Contoh

Pilih Buah:

Pilih Warna Favorit:

- ☒ Merah
- ☒ Biru
- ☒ Hijau

Pilih Jenis Kelamin:

- ☒ Laki-laki
- ☐ Perempuan

Anda memilih buah: apel

Warna favorit anda: merah, biru, hijau

Jenis kelamin Anda: laki-laki

This code is an example of an HTML form that uses jQuery to send data asynchronously to the server. The user can select a type of fruit, favorite color, and gender, then click the "Submit" button. When the button is pressed, the data from the form will be collected and sent to the file `proses_lanjut.php` using the POST method without refreshing the page. After the data is sent, the response from the server will be displayed in a div with the id "results", allowing the user to see the results directly on the same page.

No17:

```

<!DOCTYPE html>
<html>
    <head>
        <title>Form Input dengan Validasi</title>
    </head>
    <body>
        <h1>Form Input dengan Validasi</h1>
        <form method="post" action="proses_validasi.php">
            <label for="nama">Nama:</label>
            <input type="text" id="nama" name="nama">
            <br>
            <label for="email">Email:</label>
            <input type="text" id="email" name="email">
            <br>

```

```

        <input type="submit" value="Submit">
    </form>
</body>
</html>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $errors = array();

    // validasi nama
    if (empty($nama)) {
        $errors[] = "Nama harus diisi.";
    }

    // validasi email
    if (empty($email)) {
        $errors[] = "Email harus diisi.";
    } else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Format email tidak valid.";
    }

    // validasi password
    if (empty($password)) {
        $errors[] = "Password harus diisi.";
    }

    // jika ada kesalahan validasi
    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        // lanjutkan dengan pemrosesan data jika semua validasi berhasil
        // misalnya, menyimpan data ke database atau mengirim email
        echo "Data berhasil dikirim: Nama = $nama, Email = $email,
Password=$password";
    }
}

```



```
?>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $errors = array();

    // validasi nama
    if (empty($nama)) {
        $errors[] = "Nama harus diisi.";
    }

    // validasi email
    if (empty($email)) {
        $errors[] = "Email harus diisi.";
    } else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Format email tidak valid.";
    }

    // jika ada kesalahan validasi
    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        // lanjutkan dengan pemrosesan data jika semua validasi berhasil
        // misalnya, menyimpan data ke database atau mengirim email
        echo "Data berhasil dikirim: Nama = $nama, Email = $email";
    }
}
?>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $errors = array();

    // validasi nama
    if (empty($nama)) {
```

```

        $errors[] = "Nama harus diisi.";
    }

    // validasi email
    if (empty($email)) {
        $errors[] = "Email harus diisi.";
    } else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Format email tidak valid.";
    }

    // jika ada kesalahan validasi
    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        // lanjutkan dengan pemrosesan data jika semua validasi berhasil
        // misalnya, menyimpan data ke database atau mengirim email
        echo "Data berhasil dikirim: Nama = $nama, Email = $email";
    }
}
?>

```

## Form Input dengan Validasi

Nama:

Email:

Data berhasil dikirim: Nama = Charellino Kalingga Sadewo, Email = charellinokalinggasadewo@gmail.com

This code is an example of an input form with validation in PHP that asks the user to fill in their name and email. After the user fills in the form and clicks the "Submit" button, the data is sent to the file proses\_validasi.php. If the name or email is not filled in or if the email format is invalid, an error message is displayed below the form. If all input is valid, the program displays a message that the data has been successfully submitted, along with the name and email entered.

No18:

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>
  <title>Form Input dengan Validasi</title>
</head>

<body>
  <h1>Form Input dengan Validasi</h1>
  <form id="myForm" action="proses_validasi.php" method="post">
    <label for="nama">Nama:</label>
    <input type="text" id="nama" name="nama">
    <span id="nama-error" style="color: red;"></span><br>
    <br>

    <label for="email">Email:</label>
    <input type="text" id="email" name="email">
    <span id="email-error" style="color: red;"></span><br>
    <br>

    <input type="submit" value="Submit">
  </form>
  <script>
    $(document).ready(function () {
      $("#myForm").submit(function (event) {
        var nama = $("#nama").val();
        var email = $("#email").val();
        var valid = true;

        if (nama == "") {
          $("#nama-error").text("Nama harus diisi!");
          valid = false;
        } else {
          $("#nama-error").text("");
        }
        if (email === "") {
          $("#email-error").text("Email harus diisi!");
          valid = false;
        } else {

```

```

        $("#email-error").text("");
    }

    if (!valid) {
        event.preventDefault();
    }

    });
});
</script>
</body>

```

### Form Input dengan Validasi

Nama:

Email:

Nama harus diisi.

Email harus diisi.

Email harus diisi.

This code is an input form with validation using jQuery. When the user fills out the form and clicks the "Submit" button, the script checks if the name and email fields are filled in. If the name or email fields are empty, an error message is displayed next to the corresponding field, and the form submission is aborted. If all input is valid, the form is sent to the `validation_process.php` file for further processing.

No19:

```

if (!valid) {

    $.ajax({ url: "proses_validasi.php",
        type: "POST",
        data: $(this).serialize(),

        success: function(response) {
            $("#result").html(response);
        },
        error: function() {
            $("#result").html("Terjad kesalahan saat mengirim
data.");
        }
    });
}

```

This code is an HTML form with validation using jQuery and data submission using AJAX. When a user fills out the form and clicks the "Submit" button, JavaScript will check that the name and email are complete. If any are blank or the email format is invalid, an error message will be displayed next to the relevant input fields, and the submission will be aborted. If all the

data is valid, the information from the form will be sent to the process\_validation.php file, and the server response will be displayed in a div with the ID result, without refreshing the page.

No20:

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

<head>
  <title>Form Input dengan Validasi</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>

<body>
  <h1>Form Input dengan Validasi</h1>
  <form id="myForm" action="proses_validasi.php" method="post">
    <label for="nama">Nama:</label>
    <input type="text" id="nama" name="nama">
    <span id="nama-error" style="color: red;"></span><br>
    <br>

    <label for="email">Email:</label>
    <input type="text" id="email" name="email">
    <span id="email-error" style="color: red;"></span><br>
    <br>

    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
    <span id="password-error" style="color: red;"></span><br>
    <br>

    <input type="submit" value="Submit">
  </form>
  <script>
    $(document).ready(function () {
      $("#myForm").submit(function (event) {
        var nama = $("#nama").val();
        var email = $("#email").val();
        var password = $("#password").val();
        var valid = true;

        if (nama == "") {
```

```

        $("#nama-error").text("Nama harus diisi!");
        valid = false;
    } else {
        $("#nama-error").text("");
    }
    if (email === "") {
        $("#email-error").text("Email harus diisi!");
        valid = false;
    } else {
        $("#email-error").text("");
    }

    if (password.length <= 8) {
        $("#password-error").text("Password minimal 8
karakter!");
        valid = false;
    } else {
        $("#password-error").text("");
    }

    if (!valid) {
        $.ajax({ url: "proses_validasi.php",
            type: "POST",
            data: $(this).serialize(),

            success: function(response) {
                $("#result").html(response);
            },
            error: function() {
                $("#result").html("Terjad kesalahan saat mengirim
data.");
            }
        });
    }
});
});
</script>
</body>

```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST["nama"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $errors = array();

    // validasi nama
    if (empty($nama)) {
        $errors[] = "Nama harus diisi.";
    }

    // validasi email
    if (empty($email)) {
        $errors[] = "Email harus diisi.";
    } else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Format email tidak valid.";
    }

    // validasi password
    if (empty($password)) {
        $errors[] = "Password harus diisi.";
    }

    // jika ada kesalahan validasi
    if (!empty($errors)) {
        foreach ($errors as $error) {
            echo $error . "<br>";
        }
    } else {
        // lanjutkan dengan pemrosesan data jika semua validasi berhasil
        // misalnya, menyimpan data ke database atau mengirim email
        echo "Data berhasil dikirim: Nama = $nama, Email = $email,
Password=$password";
    }
}
?>
```

## Form Input dengan Validasi

Nama:  Nama harus diisi!

Email:  Email harus diisi!

Password:  Password minimal 8 karakter!

## Form Input dengan Validasi

Nama:  Nama harus diisi!

Email:

Password:

## Form Input dengan Validasi

Nama:

Email:  Email harus diisi!

Password:

## Form Input dengan Validasi

Nama:

Email:

Password:  Password minimal 8 karakter!

## Form Input dengan Validasi

Nama:

Email:

Password:

Data berhasil dikirim: Nama = Charel, Email = charellinokalinggasadewo@gmail.com, Password=admin 123

The code above is an HTML form equipped with validation using jQuery and PHP to input name, email, and password. When the user tries to submit the form, jQuery will check if all input is valid: name cannot be empty, email must have the correct format, and password must be at least 8 characters long. If there are any errors, an error message will be displayed below the relevant input field, and the form submission will be stopped. If all input is valid, the data will be sent to the file proses\_validasi.php using AJAX, and the result from the server (either an error message or confirmation) will be displayed below the form.