

Validation Form

Keamanan Perangkat Lunak



Disusun oleh:

Figo Gymnastiar Farhaan Pratama

20051397015

**PROGRAM STUDI MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA
2022**

File PHP dan SQL dapat diakses melalui tautan Github berikut.
<https://github.com/figopratama/kpl.git>

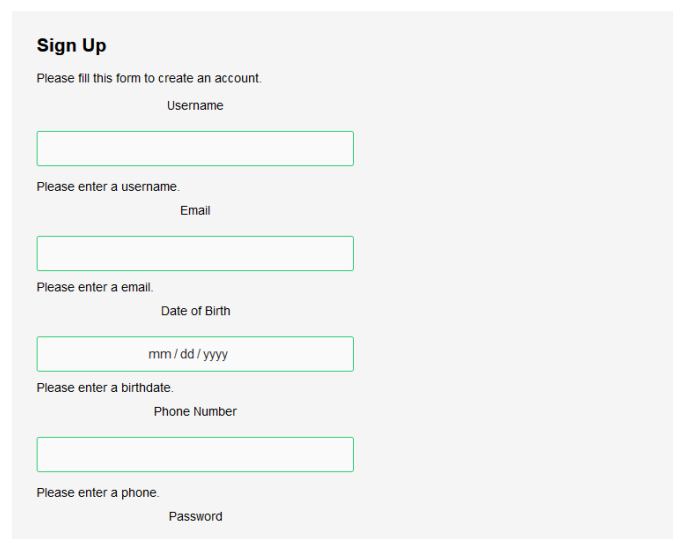
I. Kode, Penjelasan, dan Tampilan Antarmuka

A. Required Input

Fitur ini bertujuan agar user diharuskan untuk memberikan input data yang terdapat program “empty” pada formulirnya.

```
if(empty(trim($_POST["username"]))) {  
    $username_err = "Please enter a username.";  
if(empty(trim($_POST["email"]))) {  
    $email_err = "Please enter a email.";  
if(empty(trim($_POST["birthdate"]))) {  
    $birthdate_err = "Please enter a birthdate.";  
if(empty(trim($_POST["phone"]))) {  
    $phone_err = "Please enter a phone.";  
if(empty(trim($_POST["password"]))) {  
    $password_err = "Please enter a password.";
```

Output:



The screenshot shows a 'Sign Up' form with the following fields and error messages:

- Username:** Please enter a username.
- Email:** Please enter a email.
- Date of Birth:** Please enter a birthdate.
- Phone Number:** Please enter a phone.
- Password:** Please enter a password.

B. Pembatasan Isian Data

Pembatasan data digunakan agar user tidak dapat menginput data yang tidak dibutuhkan oleh tiap kategori form. Contoh pada program, pembatasan karakter diberikan pada function preg_match variabel pertama.

```
elseif(!preg_match('/^[a-zA-Z0-9_]+$/', trim($_POST["username"]))) {  
    $username_err = "Username can only contain letters, numbers, and  
underscores.";
```

```
elseif(!preg_match('/^[a-zA-Z0-9@.]+$/ ', trim($_POST["email"]))) {
    $email_err = "email can only contain letters, numbers, and underscores.";
}
elseif(!preg_match('/^[0-9-]+$/ ', trim($_POST["birthdate"]))) {
    $birthdate_err = "birthdate can only contain a date.";
}
elseif(!preg_match('/^[0]{1}[0-9]{3}[0-9]{4}[0-9]{4}$/ ', trim($_POST["phone"]))) {
    $phone_err = "Phone number can only contain numbers";
}
```

Output:

The image shows two versions of a 'Sign Up' form. The left version shows the form with filled-in fields: Username 'admin1?/=', Email 'admin1@gmail_.com=', Date of Birth '10 / 08 / 1987', and Phone Number '1'. The right version shows the form with empty fields and placeholder text: Username, Email, Date of Birth 'mm / dd / yyyy', and Phone Number. Both versions include error messages below the fields.

C. DateTime Ticker

DateTime Ticker digunakan untuk mengambil data user berupa tanggal dan waktu, seperti contoh pada program ini yaitu user mengisi form tanggal lahir saat melakukan registrasi. Pada program ini yaitu dimana form memiliki tipe “date” sehingga input yang diterima juga hanya berupa tanggal dengan format mm/dd/yyyy.

```
<div class="form-group">
    <label>Date of Birth</label>
    <input type="date" name="birthdate"
        class="form-control <?php echo (!empty($birthdate_err)) ? 'is-invalid' :
        '' ; ?>" value="<?php echo $birthdate; ?>">
    <span class="invalid-feedback"><?php echo $birthdate_err; ?></span>
</div>
```

Output:

Email

Date of Birth

mm/dd/yyyy

September 2022

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Submit

Reset

Already have an account? [Login here](#)

D. Validasi Email

Berguna untuk memeriksa apakah email yang didaftarkan user di laman Registrasi.php adalah belum terdaftar pada database. Jika iya (bernilai 0), maka program akan menampilkan pesan kesalahan bahwa email sudah terdaftar. Sebaliknya jika belum terdaftar (bernilai 1), maka email akan berhasil didaftarkan pada database.

// Fungsi SQL untuk memilih email yang akan divalidasi

```
$sql = "SELECT id FROM users WHERE email = ?";
```

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

// Nyatakan variabel sebagai statement yang sudah didefinisikan sebagai parameters

```
mysqli_stmt_bind_param($stmt, "s", $param_email);
```

// Mendefinisikan parameters

```
$param_email = trim($_POST["email"]);
```

// Query untuk mengeksekusi statement yang sudah didefinisikan

```
if(mysqli_stmt_execute($stmt)){
```

// Memindahkan hasil eksekusi dari statement yang sudah didefinisikan

```
mysqli_stmt_store_result($stmt);
```

// Memeriksa apakah email terdaftar pada database.

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

```

        $email_err = "This email is already taken."; // Jika email
sudah terdaftar, tampilkan pesan kesalahan
    } else{
        $email = trim($_POST["email"]); // Jika email belum terdaftar,
maka email berhasil dipilih
    }
} else{
    // Jika proses registrasi gagal, tampilkan pesan kesalahan
    echo "Oops! Something went wrong. Please try again later.";
}
// Close statement
mysqli_stmt_close($stmt);
}

```

E. Validasi Nomor Telepon

Sama fungsinya seperti pada validasi email, program ini dibuat untuk memvalidasi nomor telepon user yang akan didaftarkan pada database.

```

// Fungsi SQL untuk memilih nomor telepon yang akan divalidasi
$sql = "SELECT id FROM users WHERE phone = ?";

if($stmt = mysqli_prepare($link, $sql)){
    // Nyatakan variabel sebagai statement yang sudah didefinisikan
sebagai parameters

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

    // Mendefinisikan parameters

    $param_phone = trim($_POST["phone"]);

    // Query untuk mengeksekusi statement yang sudah didefinisikan

    if(mysqli_stmt_execute($stmt)){
        // Memindahkan hasil eksekusi dari statement yang sudah
didefinisikan

        mysqli_stmt_store_result($stmt);

        // Proses validasi nomor telepon.

```

```

        if(mysqli_stmt_num_rows($stmt) == 1){
            $email_err = "This phone number is already taken."; // Jika
            nomor telepon sudah terdaftar, tampilkan pesan kesalahan
        } else{
            $email = trim($_POST["phone"]); // Jika nomor telepon belum
            terdaftar, maka email berhasil dipilih
        }
    } else{
        // Jika proses registrasi gagal, tampilkan pesan kesalahan
        echo "Oops! Something went wrong. Please try again later.";
    }
    // Close statement
    mysqli_stmt_close($stmt);
}

```

F. Numeric-only Input

Numeric-only Input berguna untuk user agar tidak dapat memberi inputan selain angka. Contoh programnya seperti pada kode dibawah yaitu menggunakan function `this.value.replace` dan `replace`. Program ini umumnya digunakan untuk mengisi form kategori nomor telepon, umur, dsb.

```

<div class="form-group">
    <label>Phone Number</label>
    <input type="text" name="phone" minlength="13" oninput="this.value =
    this.value.replace(/[^0-9.]/g, '').replace(/(\..*)\./g, '$1');"
    class="form-control">?php echo (!empty($phone_err)) ? 'is-invalid' : '';
?>" value="<?php echo $phone; ?>">
    <ul class="input-requirements">
        <li>Maximum 13 characters long</li>
        <li>Must only contain numbers</li>
    </ul>
    <span class="invalid-feedback"><?php echo $phone_err; ?></span>
</div>

```

G. Realtime Validation

Realtime validation digunakan agar dapat membantu user mengisi form. Karena tugasnya adalah memberikan pesan kesalahan secara langsung kepada user selama menginputkan data pada laman Registrasi.php.

```
<div class="form-group">
    <label>Username</label>
    <input type="text" name="username" minlength="3"
        class="form-control <?php echo (!empty($username_err)) ? 'is-invalid'
: '' ; ?>" value="<?php echo $username; ?>"
        <ul class="input-requirements">
            <li>At least 3 characters long</li>
            <li>Must only contain letters and numbers</li>
        </ul>
        <span class="invalid-feedback"><?php echo $username_err; ?></span>
</div>
```

```
<div class="form-group">
    <label>Phone Number</label>
    <input type="text" name="phone" minlength="13" oninput="this.value =
this.value.replace(/[^\0-9.]/g, '').replace(/(\..*)\./g, '$1');"
        class="form-control<?php echo (!empty($phone_err)) ? 'is-invalid' :
'' ; ?>" value="<?php echo $phone; ?>"
        <ul class="input-requirements">
            <li>Maximum 13 characters long</li>
            <li>Must only contain numbers</li>
        </ul>
        <span class="invalid-feedback"><?php echo $phone_err; ?></span>
</div>
```

```
<div class="form-group">
    <label>Password</label>
    <input type="password" name="password"
        class="form-control <?php echo (!empty($password_err)) ? 'is-invalid'
: '' ; ?>" value="<?php echo $password; ?>"
        <ul class="input-requirements">
            <li>At least 8 characters long (and less than 100
characters)</li>
            <li>Contains at least 1 number</li>
            <li>Contains at least 1 lowercase letter</li>
```

```

        <li>Contains at least 1 uppercase letter</li>
        <li>Contains a special character (e.g. @ !)</li>
    </ul>

    <span class="invalid-feedback"><?php echo $password_err; ?></span>
</div>

```

Output:

Date of Birth

Phone Number

- Maximum 13 characters long
- Must only contain numbers

Password

- At least 8 characters long (and less than 100 characters)
- Contains at least 1 number
- Contains at least 1 lowercase letter
- Contains at least 1 uppercase letter
- Contains a special character (e.g. @ !)

Confirm Password