



## **Modul 3. Implementasi Antar Muka**

### **3.1 Pendahuluan**

Pada umumnya aplikasi yang dibuat untuk transaksional data memiliki fungsi input data. Input data tersebut seringkali menjadi celah kesalahan pada aplikasi. Pada praktikum ini berisi implementasi antar muka yang dikhususkan pada pengecekan data input pada interface meliputi: Existence, Data-Type, Domain, dan Format checks. Implementasi dilakukan dengan html5 dan javascript.

#### **3.1.1 Existence checks**

Existence checks merupakan pengujian atau validasi apakah sudah ada nilai pada suatu input atau tidak. Pengecekan ini sangat penting dilakukan untuk input data yang mandatory atau tidak boleh kosong.

#### **3.1.2 Data-type checks**

Data-type checks atau pengecekan tipe data berfungsi untuk memvalidasi tipe data yang sesuai dengan diinputkan. Sebagai contoh adalah isian untuk nomor induk, telp, dsb itu tidak boleh diisi oleh huruf.

#### **3.1.3 Domain Checks**

Domain checks atau range dan constraint check dapat memeriksa input pengguna untuk konsistensi dengan rentang minimum / maksimum, atau konsistensi dengan tes untuk mengevaluasi urutan karakter, seperti satu atau lebih tes terhadap ekspresi reguler. Misalnya, NIM harus memiliki 10 digit atau jumlah pembelian tidak boleh kurang dari sama dengan 0.

#### **3.1.4 Format checking**

Format check dapat memeriksa input yang diberikan kepada sistem apakah input tersebut mempunyai format yang sesuai dengan yang dibutuhkan. Format check yang sering digunakan adalah format email, format tanggal, angka, kode, dsb. Penggunaan format check berkaitan erat dengan penggunaan Regular Expression (Regex).

Regex yang digunakan untuk mengecek format email adalah

```
/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3})+$/
```

Penjelasan Regex untuk format email diatas

| Karakter      | Deskripsi   |
|---------------|---|
| / .. /        | Semua RegEx dimulai dan diakhiri dengan garis miring ke depan   |
| ^             | Menandakan awal string atau garis.  |
| \w+           | Menandakan satu atau lebih karakter kata termasuk garis bawah. Setara dengan [A-Za-z0-9_]   |
| [\. -]        | \ Menunjukkan bahwa karakter berikutnya adalah karakter spesial.<br>.- Menunjukan cocok dengan karakter . atau -  |
| ?             | Menandakan adanya karakter sebelumnya 0 atau 1 kali. Di sini karakter sebelumnya adalah [.]   |
| \w+           | Menandakan 1 atau lebih karakter kata termasuk garis bawah. Setara dengan [A-Za-z0-9_]  |
| *             | Menunjukan karakter sebelumnya dapat 0 kali atau lebih  |
| ([.]?\w+)*    | menunjukan format [.]? \ w + dapat terjadi 0 kali atau lebih  |
| \w+([.]?\w+)* | Sub-ekspresi \ w + ([.]? \ w +) * digunakan untuk mencocokkan nama pengguna dalam email. Ini dimulai dengan setidaknya satu karakter kata atau lebih termasuk garis bawah, setara dengan [A-Za-z0-9_]. , diikuti oleh . atau dan . atau - harus diikuti oleh karakter kata (A-Za-z0-9_) |
| @             | Menandakan karakter @.  |
| \w+([.]?\w+)* | Sub-ekspresi \ w + ([.]? \ w +) * digunakan untuk mencocokkan nama pengguna dalam email. Ini dimulai dengan setidaknya satu karakter kata atau lebih termasuk garis bawah, setara dengan [A-Za-z0-9_]. , diikuti oleh . atau dan . atau - harus diikuti oleh karakter kata (A-Za-z0-9_) |
| \\.w{2,3}     | Cocok dengan tanda . diikuti oleh dua atau tiga karakter kata   |
| +             | Tanda + menentukan bahwa sub-ekspresi di atas akan muncul satu kali atau lebih, mis., .Com, .co.us, .edu.uk dll.  |
| \$            | menandakan akhir string atau garis  |

### 3.2 Latihan

Buatlah folder MODUL 3 dalam folder repository yang terkoneksi ke github (Modul 1). Dalam folder tersebut, buatlah code, berdasarkan latihan-latihan berikut dengan code editor



yang disukai dan dikuasai, tambahkan signature berupa code komentar berisi NIM NAMA dan KELAS. Kemudian coba dijalankan code tersebut dalam browser. Uji menggunakan input yang sesuai. Setiap output harap di screenshot, karena akan dijadikan delivery pengerjaan praktikum (screenshot + source code). Petunjuk pengumpulan hasil praktikum ada pada sub bab terakhir.

### 3.2.1 Existence checks

```
<!DOCTYPE html>
<html>
<head>
    <title>Existence Checks</title>
</head>
<body>
<form name="myForm" action="" onsubmit="return validateForm()"
method="POST">
<input type="text" id="username" name="username">
<input type="submit" value="Submit">
</form>
<script type="text/javascript">
    function validateForm(){
        var user = document.myForm.username.value;
        if(user == ""){
            alert("Username tidak boleh kosong");
            return false;
        }else{
            return true;
        }
    }
</script>
</body>
</html>
```

### 3.2.2 Data-type checks

```
<!DOCTYPE html>
<html>
<head>
    <title>Data-Type Checks</title>
</head>
<body>
<form name="myForm" action="" onsubmit="return validateForm()"
method="POST">
<input type="text" id="telp" name="telp">
```



```
<input type="submit" value="Submit">
</form>
<script type="text/javascript">
    function validateForm(){
        var telp = document.myForm.telp.value;
        if(isNaN(telp)){
            alert("Telp harus berupa angka");
            return false;
        }else{
            return true;
        }
    }
</script>
</body>
</html>
```

isNaN merupakan kepanjangan dari **“is Not a Number”** jika variable bukan number maka akan mengembalikan nilai true, sebaliknya akan false.

### 3.2.3 Domain Checks

```
<!DOCTYPE html>
<html>
<head>
    <title>Domain Checks</title>
</head>
<body>
<form name="myForm" action="" onsubmit="return validateForm()"
method="POST">
NIM <input type="text" id="nim" name="name"> NIM harus 10 digit.<br>
Semester <input type="text" id="sem" name="sem"> Semester tidak boleh
kurang dari 1 dan lebih dari 12 <br>
<input type="submit" value="Submit">
</form>
<script type="text/javascript">
    function validateForm(){
        var nim = document.myForm.nim.value;
        var sem = document.myForm.sem.value;
        if(nim.length != 10){
            alert("NIM harus 10 digit");
            return false;
        }
        if((sem<1) || (sem>12)){
```



```
        alert("Semester harus diisi oleh 1-12");
        return false;
    }
    return true;
}
</script>
</body>
</html>
```

### 3.2.4 Format checking

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>JavaScript form validation - checking email</title>
    <style type="text/css">
        li {
            list-style-type: none;
            font-size: 16pt;
        }
    </style>
</head>
<body>
    <h2>Input email dan Submit</h2>
    <form name="form1" action="#">
        <ul>
            <li><input type='text' name='text1' /></li>
            <li>&nbsp;</li>
            <li class="submit"><input type="submit" name="submit"
value="Submit"
onclick="ValidateEmail(document.form1.text1)" /></li>
        </ul>
    </form>
</div>
<script type="text/javascript">
    function ValidateEmail(inputText)
    {
var mailformat = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3})+$/;
if(inputText.value.match(mailformat))
    {
document.form1.text1.focus();
```



```
return true;
}
else
{
    alert("Alamat email tidak valid!");
    document.form1.text1.focus();
    return false;
}
}
</script>
</body>

</html>
```

### 3.3 Tugas

Buatlah validasi untuk form berikut :

```
<!DOCTYPE html>
<html>
<head>
    <title>Tugas Implementasi Interface</title>
</head>
<body>
    <form name="formTugas" id="tugasku">
        <input type="text" name="username" id="username"
placeholder="Username" ><br>
        <input type="password" name="password" id="pass"
placeholder="Password"><br>
        <input type="password" name="repeat" id="repeat" placeholder="Ulangi
Password"><br>
        <input type="text" name="email" id="email" placeholder="Email"><br>
        <input type="text" name="telp" id="telp" placeholder="Nomor
Telpon"><br>
        <textarea id="alamat" style="width: 167px;">Alamat</textarea><br>
        <input type="submit" name="" value="Submit">
    </form>
</body>
</html>
```

Dengan aturan sebagai berikut :

- Username 3-15 karakter
- Password min 4 digit harus ada Huruf besar Symbol dan Angka
- Ulangi Password harus sama dengan Password

- email format email
- Nomor Telepon harus 10 - 12 digit
- Alamat tidak boleh kosong

### **Petunjuk Pengumpulan Hasil Praktikum**

1. Nama file gunakan format: **MODUL3<SPASI>NIM<SPASI>NAMA.**
2. Dokumentasi Praktikum (.doc, .pdf):
  - a. Cover (Sesuai format laporan praktikum)
  - b. Latihan (Screenshot output latihan-latihan)
  - c. Tugas
    - i. Source code lengkap (Gunakan 2 kolom, copy paste dari editor, atau menggunakan font CONSOLAS ukuran 10)
    - ii. Screenshot output Tugas
  - d. Kesimpulan (Uraian berupa narasi atau deskripsi tentang pengamatan hasil kegiatan praktikum)
3. Source code Hasil Praktikum
  - a. PUSH ke dalam repository github yang digunakan pada modul 1
  - b. Archive (.rar, .zip, .dll) semua sourcecode yang digunakan.

File dokumentasi dan source code tidak boleh digabung (.rar, .zip) kumpulkan secara terpisah.