

Setoran:

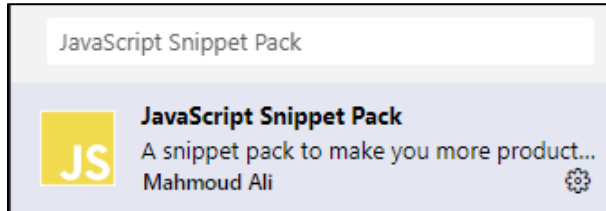
File Laporan Pengamatan: **PABWE_LaporanPraktikum5_{NIM}.pdf**

Source Code: **PABWE_SourceCode5_{NIM}.zip**

Persiapan Praktikum:

1. Extensions for VSCode

- a) Install extension dengan nama "JavaScript Snippet Pack".



2. Pemahaman JavaScript

Silahkan belajar secara mandiri terkait JavaScript Dasar dan JavaScript DOM.

Latihan Praktikum:

Praktikan, amati, pahami, dan jawab semua pertanyaan yang ada serta laporkan hasilnya pada laporan praktikum.

(!) Selesaikan terlebih dahulu kode program untuk dapat menjawab pertanyaan.

A. Form Validation

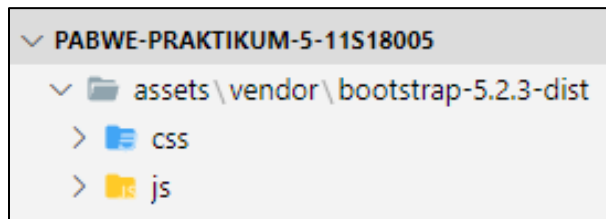
Validasi formulir adalah proses memeriksa data yang dimasukkan oleh pengguna dalam formulir web untuk memastikan bahwa data tersebut memenuhi persyaratan yang ditentukan sebelum data tersebut dikirimkan atau diproses lebih lanjut.

| Workspace

Buat workspace baru pada VSCode dengan penamaan "**pabwe-praktikum-5-{nim}**".

| Assets

Download dan extract assets dari Delcom ke workspace kamu, seperti berikut:



1. Validasi Setelah Menekan Tombol Submit

Buat dan modifikasi isi file "**form1.html**", seperti berikut:

```
form1. html
1 <!DOCTYPE html >
2 <html lang="en">
3 <head>
```

form1.html

```
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial -
scale=1.0" />
6      <title>Form Validasi </title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10             background-color: #f4f4f4;
11             margin: 0;
12             padding: 0;
13         }
14
15         h1,
16         h3 {
17             text-align: center;
18         }
19
20         h1 {
21             margin-bottom: 0;
22         }
23
24         h3 {
25             margin-top: 0;
26             color: #808080;
27         }
28
29         form {
30             max-width: 400px;
31             margin: 0 auto;
32             background: #fff;
33             padding: 20px;
34             border-radius: 5px;
35             box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
36         }
37
38         label {
39             display: block;
40             font-weight: bold;
41             margin-bottom: 5px;
42         }
43
44         input[type="text"],
45         input[type="email"],
46         select {
47             padding: 10px;
48             border: 1px solid #ccc;
49             border-radius: 5px;
50         }
51
52         input[type="text"],
```

form1.html

```

53     input[type="email"] {
54         width: calc(100% - 20px);
55     }
56
57     select {
58         width: 100%;
59     }
60
61     .error {
62         color: red;
63     }
64
65     input[type="submit"] {
66         background-color: #007bff;
67         color: #fff;
68         padding: 10px 20px;
69         border: none;
70         border-radius: 5px;
71         cursor: pointer;
72     }
73
74     input[type="submit"]:hover {
75         background-color: #0056b3;
76     }
77
78     .mb-15 {
79         margin-bottom: 15px;
80     }
81 </style>
82 </head>
83 <body>
84     <h1>Form Validasi</h1>
85     <h3>Validasi setelah menekan tombol Submit</h3>
86     <form id="myForm" onsubmit="return validateForm()">
87         <div class="mb-15">
88             <label for="nama">Nama: </label>
89             <input type="text" id="nama" name="nama" />
90             <span id="namaError" class="error"></span>
91         </div>
92
93         <div class="mb-15">
94             <label for="email">Email: </label>
95             <input type="email" id="email" name="email" />
96             <span id="emailError" class="error"></span>
97         </div>
98
99         <div class="mb-15">
100             <label for="role">Role: </label>
101             <select id="role" name="role">
102                 <option value="">~Pilih Role</option>

```

form1. html

```

103         <opti on>Pengaj ar</opti on>
104         <opti on>Pel aj ar</opti on>
105     </sel ect>
106     <span id="rol eError" cl ass="error"></span>
107 </di v>
108
109     <di v cl ass="mb-15">
110         <input type="submi t" val ue="Submi t" />
111     </di v>
112 </form>
113
114 <scri pt>
115     functi on vali dateForm() {
116         let status = true;
117
118         var nama = document. getEl ementByI d("nama"). val ue. trim();
119         var email = document. getEl ementByI d("email"). val ue. trim();
120         var role = document. getEl ementByI d("role"). val ue. trim();
121
122         var namaError = document. getEl ementByI d("namaError");
123         var emailError = document. getEl ementByI d("emailError");
124         var roleError = document. getEl ementByI d("roleError");
125
126         namaError. innerHTML = "";
127         emailError. innerHTML = "";
128         roleError. innerHTML = "";
129
130         if (nama === "") {
131             namaError. innerHTML = "Nama harus di isi!";
132             status = false;
133         } else if (nama. length < 8) {
134             namaError. innerHTML = "Panj ang nama mi nimal 8 karakter!";
135             status = false;
136         } else {
137             const regex = /^[A-Za-z\s]*$/;
138             if (!regex. test(nama)) {
139                 namaError. innerHTML =
140                     "Nama hanya boleh terdiri dari huruf al fabet dan spasi!";
141                 status = false;
142             }
143         }
144
145         if (email === "") {
146             emailError. innerHTML = "Email harus di isi!";
147             status = false;
148         } else {
149             const emailPattern =
150                 /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
151
152             if (!emailPattern. test(email)) {

```

```
form1.html
153         emailError.innerHTML = "Format email tidak valid!";
154         status = false;
155     }
156 }
157
158 if (role === "") {
159     roleError.innerHTML = "Role harus dipilih!";
160     status = false;
161 }
162
163 return status;
164 }
165 </script>
166 </body>
167 </html>
```

Analisis cara kerja kode program di atas. Jalankan aplikasi menggunakan Live Server, selanjutnya amati tampilan pada halaman web browser dan console web browser.

2. Validasi dengan Toggle Class

Buat dan modifikasi isi file **“form2.html”**, seperti berikut:

```
form2.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-
  scale=1.0" />
6 <title>Form Validasi</title>
7
8 <link
9   rel="stylesheet"
10  href="assets/vendor/bootstrap-5.2.3-dist/css/bootstrap.min.css"
11 />
12 </head>
13
14 <body>
15 <div class="mx-auto mt-3" style="max-width: 400px">
16 <h1 class="text-center mb-0">Form Validasi</h1>
17 <h3 class="text-center text-muted">Validasi dengan Toggle
  Class</h3>
18
19 <div class="card">
20 <div class="card-body">
21 <form
22   id="myForm"
23   class="needs-validation"
24   onsubmit="return validateForm()"
25 >
26 <div class="mb-3">
```

form2. html

```

27         <label class="form-label" for="nama">Nama</label>
28         <input class="form-control" type="text" id="nama"
name="nama" />
29         <div class="invalid-feedback" id="namaError">Error</div>
30     </div>
31
32     <div class="mb-3">
33         <label class="form-label" for="email">Email</label>
34         <input
35             class="form-control"
36             type="email"
37             id="email"
38             name="email"
39         />
40         <div class="invalid-feedback" id="emailError">Error</div>
41     </div>
42
43     <div class="mb-3">
44         <label for="role">Role</label>
45         <select class="form-select" id="role" name="role">
46             <option value="">~Pilih Role~</option>
47             <option>Pengajar</option>
48             <option>Pelajar</option>
49         </select>
50         <div class="invalid-feedback" id="roleError">Error</div>
51     </div>
52
53     <div class="mb-3 text-end">
54         <input class="btn btn-primary" type="submit"
value="Submit" />
55     </div>
56 </form>
57 </div>
58 </div>
59 </div>
60
61 <script>
62     function validateForm() {
63         let status = {
64             nama: true,
65             email: true,
66             role: true,
67         };
68
69         let elmntNama = document.getElementById("nama");
70         let elmntEmail = document.getElementById("email");
71         let elmntRole = document.getElementById("role");
72
73         elmntNama.classList.remove("is-invalid");
74         elmntEmail.classList.remove("is-invalid");

```

form2. html

```
75     elmntRole.classList.remove("is-invalid");
76
77     var nama = elmntNama.value.trim();
78     var email = elmntEmail.value.trim();
79     var role = elmntRole.value.trim();
80
81     var namaError = document.getElementById("namaError");
82     var emailError = document.getElementById("emailError");
83     var roleError = document.getElementById("roleError");
84
85     namaError.innerHTML = "";
86     emailError.innerHTML = "";
87     roleError.innerHTML = "";
88
89     if (nama === "") {
90         namaError.innerHTML = "Nama harus diisi!";
91         status.nama = false;
92     } else if (nama.length < 8) {
93         namaError.innerHTML = "Panjang nama minimal 8 karakter!";
94         status.nama = false;
95     } else {
96         const regex = /^[A-Za-z\s]*$/;
97         if (!regex.test(nama)) {
98             namaError.innerHTML =
99                 "Nama hanya boleh terdiri dari huruf alfabet dan spasi!";
100             status.nama = false;
101         }
102     }
103
104     if (email === "") {
105         emailError.innerHTML = "Email harus diisi!";
106         status.email = false;
107     } else {
108         const emailPattern =
109             /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
110
111         if (!emailPattern.test(email)) {
112             emailError.innerHTML = "Format email tidak valid!";
113             status.email = false;
114         }
115     }
116
117     if (role === "") {
118         roleError.innerHTML = "Role harus dipilih!";
119         status.role = false;
120     }
121
122     if (!status.nama) {
123         elmntNama.classList.add("is-invalid");
124     }
```

form2. html

```
125
126     if (!status.email) {
127         elmntEmail.classList.add("is-invalid");
128     }
129
130     if (!status.role) {
131         elmntRole.classList.add("is-invalid");
132     }
133
134     console.log(status);
135     return status.nama && status.email && status.role;
136 }
137 </script>
138 </body>
139 </html>
```

Analisis cara kerja kode program di atas. Jalankan aplikasi menggunakan Live Server, selanjutnya amati tampilan pada halaman web browser dan console web browser.

3. Validasi secara Realtime

Buat dan modifikasi isi file **“form3.html”**, seperti berikut:

form3. html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-
6       scale=1.0" />
7     <title>Form Validasi </title>
8
9     <link
10       rel="stylesheet"
11       href="assets/vendor/bootstrap-5.2.3-dist/css/bootstrap.min.css"
12     />
13   </head>
14   <body>
15     <div class="mx-auto mt-3" style="max-width: 400px">
16       <h1 class="text-center mb-0">Form Validasi </h1>
17       <h3 class="text-center text-muted">Validasi secara Real time</h3>
18
19       <div class="card">
20         <div class="card-body">
21           <form
22             id="myForm"
23             class="needs-validation"
24             onsubmit="return validateForm()"
25           >
26             <div class="mb-3">
```


form3. html

```

27         <label class="form-label "
for="inputEmail">Username</label >
28         <input
29             oninput="validateUsername(this)"
30             class="form-control "
31             type="text"
32             id="inputEmail"
33             name="username"
34             autocomplete="username"
35         />
36         <div class="invalid-feedback" id="usernameErr"></div>
37     </div>
38
39     <div class="mb-3">
40         <label class="form-label "
for="inputPassword">Password</label >
41         <input
42             oninput="validatePassword(this)"
43             class="form-control "
44             type="password"
45             id="inputPassword"
46             name="password"
47             autocomplete="current-password"
48         />
49         <ul id="passwordInfo" class="ms-1 list-unstyled">
50             <li>
51                 <span class="text-danger">
52                     &#10005; panjang minimal 8 karakter
53                 </span>
54             </li>
55             <li>
56                 <span class="text-danger">
57                     &#10005; terdapat huruf kapital
58                 </span>
59             </li>
60             <li>
61                 <span class="text-danger">
62                     &#10005; terdapat huruf kecil
63                 </span>
64             </li>
65             <li>
66                 <span class="text-danger"> &#10005; terdapat angka
67             </li>
68             <li>
69                 <span class="text-danger">
70                     &#10005; terdapat karakter spesial
71                 </span>
72             </li>
73         </ul>

```

form3. html

```

74         </div>
75
76         <div class="mb-3 text-end">
77             <input class="btn btn-primary" type="submit"
value="Submit" />
78         </div>
79     </form>
80 </div>
81 </div>
82 </div>
83
84 <script>
85     function validateUsername(targetElement) {
86         let status = true;
87         const elementErr = document.getElementById("usernameErr");
88         elementErr.innerHTML = "";
89
90         if (targetElement.value.trim() === "") {
91             elementErr.innerHTML = "Username tidak boleh kosong!";
92             status = false;
93         } else if (targetElement.value.length !== 8) {
94             elementErr.innerHTML = "Panjang username harus 8 karakter!";
95             status = false;
96         } else {
97             // example: if18005
98             const usernamePattern = /^(if[1-9][0-9]{4})$/gm;
99
100            if (!usernamePattern.test(targetElement.value)) {
101                elementErr.innerHTML =
102                    "Format username tidak mewakili prodi informatika.";
103                status = false;
104            } else if (
105                targetElement.value.slice(3, 5) >
106                new Date().getFullYear().toString().slice(2, 4)
107            ) {
108                elementErr.innerHTML = "Username dengan angkatan ini belum
tersedia.";
109                status = false;
110            }
111        }
112
113        if (status) {
114            targetElement.classList.add("is-valid");
115            targetElement.classList.remove("is-invalid");
116        } else {
117            targetElement.classList.add("is-invalid");
118            targetElement.classList.remove("is-valid");
119        }
120
121        return status;

```

form3. html

```
122     }
123
124     function validatePassword(targetElmnt) {
125         let status = true;
126         const elmntPasswordInfo =
127         document.getElementById("passwordInfo");
128         const listItems = elmntPasswordInfo.querySelectorAll("li");
129
130         if (targetElmnt.value.length < 8) {
131             listItems[0].innerHTML = `<span class="text-danger">
132                 &#10005; panjang minimal 8 karakter
133             </span>`;
134             status = false;
135         } else {
136             listItems[0].innerHTML = `<span class="text-success">
137                 &#10003; panjang minimal 8 karakter
138             </span>`;
139         }
140
141         if (!/[A-Z]/.test(targetElmnt.value)) {
142             listItems[1].innerHTML = `<span class="text-danger">
143                 &#10005; terdapat huruf kapital
144             </span>`;
145             status = false;
146         } else {
147             listItems[1].innerHTML = `<span class="text-success">
148                 &#10003; terdapat huruf kapital
149             </span>`;
150         }
151
152         if (!/[a-z]/.test(targetElmnt.value)) {
153             listItems[2].innerHTML = `<span class="text-danger">
154                 &#10005; terdapat huruf kecil
155             </span>`;
156             status = false;
157         } else {
158             listItems[2].innerHTML = `<span class="text-success">
159                 &#10003; terdapat huruf kecil
160             </span>`;
161         }
162
163         if (!/[0-9]/.test(targetElmnt.value)) {
164             listItems[3].innerHTML = `<span class="text-danger">
165                 &#10005; terdapat angka
166             </span>`;
167             status = false;
168         } else {
169             listItems[3].innerHTML = `<span class="text-success">
170                 &#10003; terdapat angka
171             </span>`;
```

```

form3. html
171     }
172
173     if (!/[!@#$$%^&*()_+{}\\[\]:;<>,.?~\\-]/.test(targetElmnt.value)
) {
174         listItems[4].innerHTML = `<span class="text-danger">
175             &#10005; terdapat karakter spesial
176             </span>`;
177         status = false;
178     } else {
179         listItems[4].innerHTML = `<span class="text-success">
180             &#10003; terdapat karakter spesial
181             </span>`;
182     }
183
184     if (status) {
185         targetElmnt.classList.add("is-valid");
186         targetElmnt.classList.remove("is-invalid");
187     } else {
188         targetElmnt.classList.add("is-invalid");
189         targetElmnt.classList.remove("is-valid");
190     }
191
192     return status;
193 }
194
195 function validateForm() {
196     const elmntUsername = document.getElementById("inputUsername");
197     const elmntPassword = document.getElementById("inputPassword");
198
199     return (
200         validateUsername(elmntUsername) &&
        validatePassword(elmntPassword)
201     );
202 }
203 </script>
204 </body>
205 </html>

```

Analisis cara kerja kode program di atas. Jalankan aplikasi menggunakan Live Server, selanjutnya amati tampilan pada halaman web browser dan console web browser.

Tantangan Praktikum:

Tantangan praktikum berisikan studi kasus yang perlu untuk diselesaikan permasalahannya agar melatih pemahaman yang telah diperoleh dari latihan praktikum.

1. Menghubungkan

Buatlah satu halaman khusus untuk menghubungkan implementasi JavaScript di atas.