

LAPORAN TUGAS 1 JAVASCRIPT



LAPORAN

Disusun untuk Memenuhi Tugas Kelompok pada Mata Kuliah Pengembangan Berbasis Platform Semester Lima yang Diampu oleh Ibu Khadijah, S.Kom., M.Cs.

Disusun Oleh :

- 1. Adiba Justinian (24060120130080)**
- 2. Melani Safwa Aprila (24060120120008)**
- 3. Ratih Adiningrum (24060120140155)**
- 4. Vania Jessica Azalia (24060120140101)**
- 5. Veronika Manalu (24060120120029)**

PROGRAM STUDI S-1 INFORMATIKA

DEPARTEMEN INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

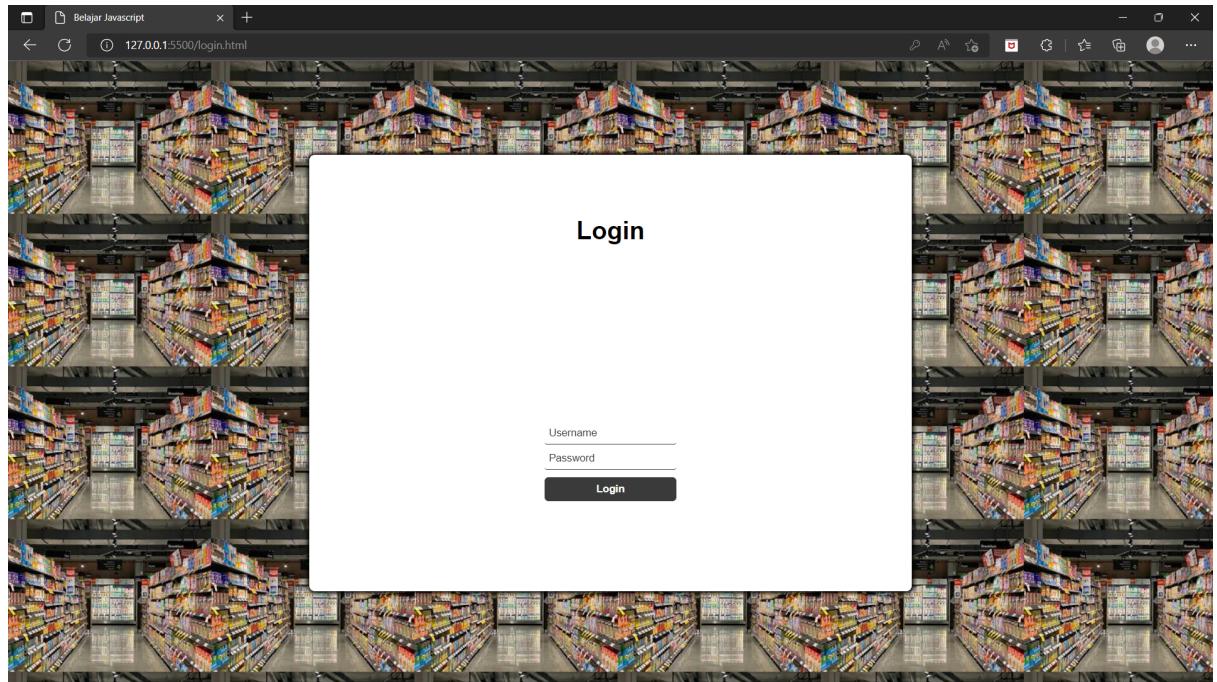
UNIVERSITAS DIPONEGORO

SEMARANG

2022

Halaman Web:

1. Tampilan awal web



Tampilan awal *website* berupa halaman *login* yang mana terdapat *form username* dan *password* serta *login button* untuk menginputkannya.

```
<main id="main-holder">
```

Tag `main` digunakan untuk mendefinisikan bagian utama suatu dokumen, yaitu kotak putih pada tampilan *form login* di atas.

```
<h1 id="login-header">Login</h1>
```

Lalu terdapat *header* atau judul utama pada tampilan di atas, yaitu tulisan “Login” yang dibuat menggunakan *heading* tag `h1`.

```
<div id="login-error-msg-holder">
  <p id="login-error-msg">Invalid username <span id="error-msg-second-line"> and/or password</span></p>
</div>
```

Kode di atas digunakan untuk membuat tampilan *error message*. Jika *username* dan/atau *password* yang diinputkan salah maka akan muncul *error message/alert* “invalid username and/or password”.

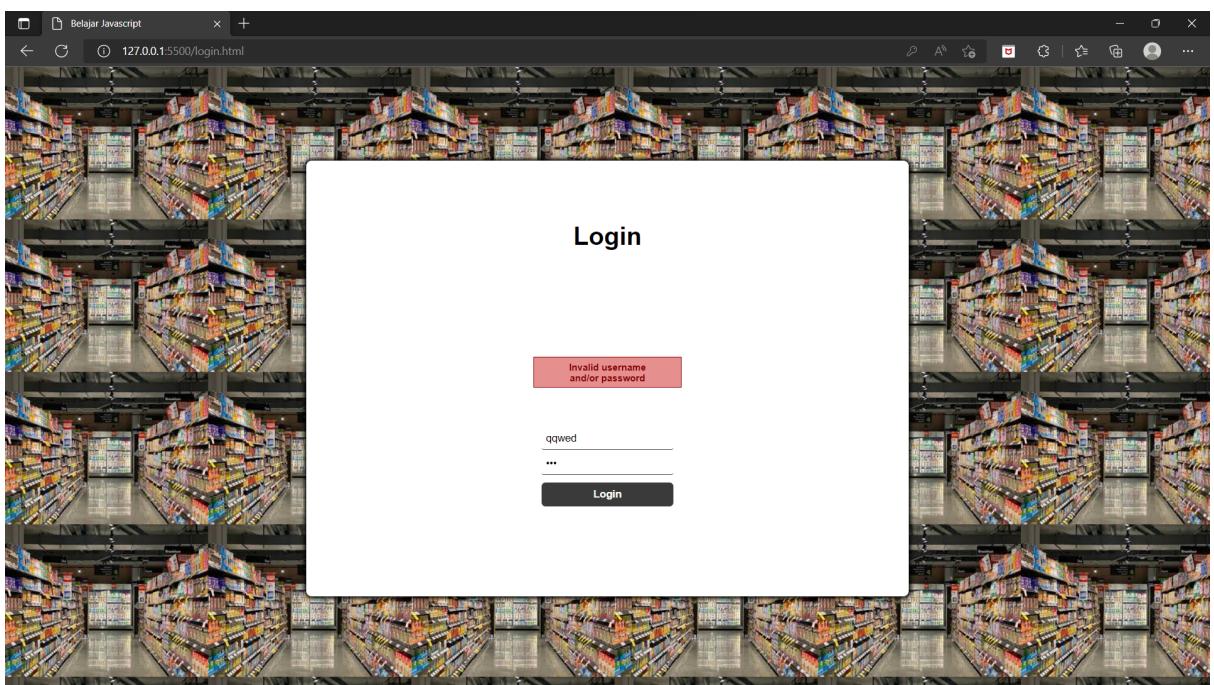
```

<form id="login-form">
    <input type="text" name="username" id="username-field"
    class="login-form-field" placeholder="Username">
    <input type="password" name="password" id="password-field"
    class="login-form-field" placeholder="Password">
    <input type="submit" value="Login" id="login-form-submit">
</form>

```

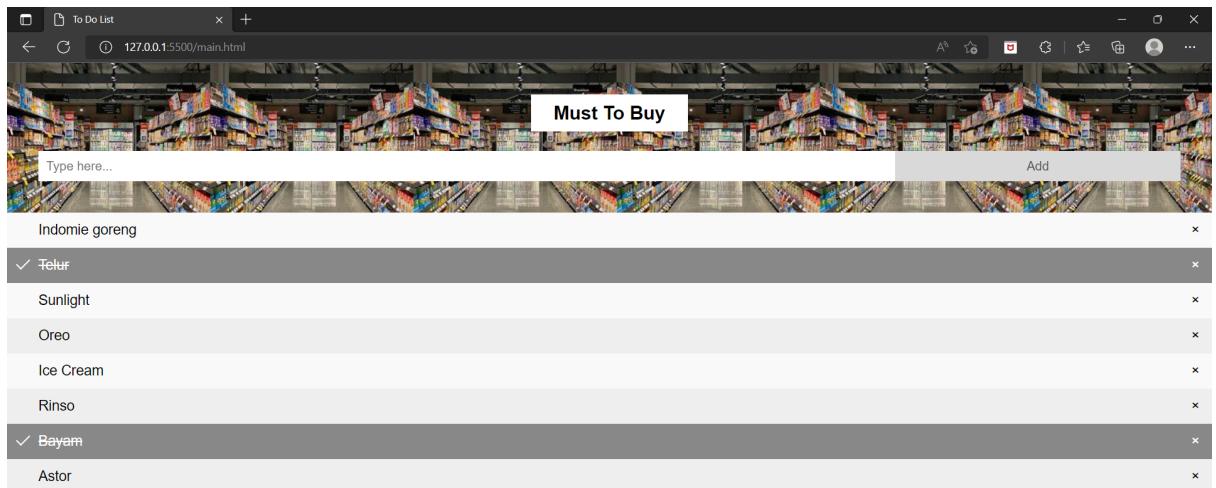
Lalu, dibuat *form field* menggunakan tag `form` dengan elemen *username* dan *password* yang dapat diisi dengan data *user* serta elemen *login* sebagai *input button*.

2. Tampilan web ketika username dan atau password salah.



Ketika memasukkan *username* dan *password* yang benar, ketika diklik tombol *login* akan berganti halaman baru menggunakan `window.location.replace("main.html")`. Tetapi jika *username* dan/ atau *password*-nya salah maka akan muncul *error message/alert* “invalid username and/or password”. Dalam hal ini kami gunakan settingan `username === "user" && password === "1"`. Kami juga menambahkan *opacity style* untuk *login error message* yaitu `loginErrorMsg.style.opacity = 1` dimana 1 berarti elemen tidak transparan.

3. Tampilan halaman baru ketika berhasil login.



Ketika berhasil *login* maka akan masuk ke halaman selanjutnya dimana *list* yang sudah maupun yang akan kita tambahkan berada.

```
<div id="myDIV" class="header">
    <h2 class="headertext">Must To Buy</h2>
    <input type="text" id="myInput" placeholder="Type here...">
    <span onclick="newElement()" class="addBtn">Add</span>
</div>
```

Headertext pada halaman ini, bertuliskan “Must To Buy” yang dibuat dengan *heading* tag h2. Lalu ditambahkan juga bidang formulir yang memungkinkan *user* untuk menginput atau mengentri data pada *list* dengan menggunakan tag *input* serta terdapat juga *button* Add dengan *event onclick* dimana *button* akan melakukan *action* ketika diklik.

```
<ul id="myUL">
    <li>Indomie goreng</li>
    <li class="checked">Telur</li>
    <li>Sunlight</li>
    <li>Oreo</li>
    <li>Ice Cream</li>
    <li>Rinso</li>
```

```
<li class="checked">Bayam</li>
<li>Astor</li>
</ul>
```

Lalu, dibuat *unordered list* menggunakan *tag* `UL`, dimana akan ditampilkan daftar *list* yang sudah dibuat pada program awal maupun yang akan ditambahkan. Penulisan elemen-elemen dalam *list* dibuat menggunakan *tag* `li` dengan `class="checked"` untuk *checked list* awal yang dibuat pada program.

Terdapat berbagai interaksi yang terjadi pada halaman web ini, diantaranya :

- Menambahkan tombol “x” pada setiap list

Membuat list barang yang diinginkan dengan menggunakan `document.getElementsByTagName("LI")` dan gunakan variabel baru yaitu `i` untuk melakukan perulangan berupa membuat tombol *close* atau “x” dengan cara membuat `classname = "close"` dan *di-append*.

```
var i;
for (i = 0; i < myNodelist.length; i++) {
    var span = document.createElement("SPAN");
    var txt = document.createTextNode("\u00D7");
    span.className = "close";
    span.appendChild(txt);
    myNodelist[i].appendChild(span);
```

- Menghapus *list* saat tombol “x” di klik

Gunakan *looping for* untuk melakukan penghapusan secara berkali-kali ketika tombol “x” atau *close* dengan cara mengubah *display* menjadi “none” :

```
for (i = 0; i < close.length; i++) {
    close[i].onclick = function() {
        var div = this.parentElement;
        div.style.display = "none";
```

- Menambahkan “checklist” pada *list* yang dipilih

Gunakan `list.addEventListener('click', function(ev{})` untuk menambahkan *checklist* (tanda *check*) pada *list* yang tersedia tepat pada *list* yang kita inginkan.

- Menambahkan *list* baru dengan tombol “add”

Gunakan `document.createElement("li")` untuk membuat elemen *list* baru dan

`document.createTextNode(inputValue)` untuk memasukkan nama barang yang diinginkan. Ketika `inputValue`/ nama barang yang dimasukkan kosong maka akan muncul notifikasi/ pemberitahuan bahwa "You must write something!". Jika tidak kosong maka `list` akan bertambah dengan nama barang kita tuliskan, gunakan `document.getElementById("myUL").appendChild(li);`

- e. Menambahkan tombol "x" pada `list` yang baru ditambahkan

Sama seperti poin a yang mana buat elemen dan teks nodenya terlebih dahulu (`document.createElement("SPAN")` dan `document.createTextNode("\u20D7")`) lalu gunakan `classname` "close" untuk membuat tampilan tombol x pada `list` yang akan ditambahkan dengan `span.appendChild(txt)` dan `li.appendChild(span)`

- f. Menghapus `list` yang baru dibuat saat tombol "x" diklik

Hampir sama dengan poin b dimana ketika tombol `close` "x" diklik akan terhapus, hanya saja kasus ini terjadi hanya pada `list` yang baru ditambahkan. Gunakan `looping for` dengan panjang sesuai dengan banyaknya `list` yang ada.

```
for (i = 0; i < close.length; i++) {  
    close[i].onclick = function() {  
        var div = this.parentElement;  
        div.style.display = "none";
```

Source Code:

1. Halaman Login

- HTML

```
<!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>Belajar Javascript</title>  
    <link rel="stylesheet" href="login.css"/>
```

```

</head>

<body>
    <main id="main-holder">
        <h1 id="login-header">Login</h1>
        <div id="login-error-msg-holder">
            <p id="login-error-msg">Invalid username <span
id="error-msg-second-line">and/or password</span></p>
        </div>

        <form id="login-form">
            <input type="text" name="username" id="username-field"
class="login-form-field" placeholder="Username">
            <input type="password" name="password"
id="password-field" class="login-form-field"
placeholder="Password">
            <input type="submit" value="Login"
id="login-form-submit">
        </form>

    </main>
    <script>
        const loginForm = document.getElementById("login-form");
        const loginButton =
document.getElementById("login-form-submit");
        const loginErrorMsg =
document.getElementById("login-error-msg");

        loginButton.addEventListener("click", (e) => {
            e.preventDefault();
            const username = loginForm.username.value;
            const password = loginForm.password.value;

            if (username === "user" && password === "1") {
                window.location.replace("main.html");
            }
            else {
                loginErrorMsg.style.opacity = 1;
            }
        })
    </script>
</body>
</html>

```

- CSS

```

html {
    height: 100%;
}

body {
    height: 100%;
    margin: 0;
}

```

```
font-family: Arial, Helvetica, sans-serif;
display: grid;
justify-items: center;
align-items: center;
background-image: url(groceries.jfif);
}

#main-holder {
width: 50%;
height: 70%;
display: grid;
justify-items: center;
align-items: center;
background-color: white;
border-radius: 7px;
box-shadow: 0px 0px 5px 2px black;
}

#login-error-msg-holder {
width: 100%;
height: 100%;
display: grid;
justify-items: center;
align-items: center;
}

#login-error-msg {
width: 23%;
text-align: center;
margin: 0;
padding: 5px;
font-size: 12px;
font-weight: bold;
color: #8a0000;
border: 1px solid #8a0000;
background-color: #e58f8f;
opacity: 0;
}

#error-msg-second-line {
display: block;
}

#login-form {
align-self: flex-start;
display: grid;
justify-items: center;
align-items: center;
}

.login-form-field::placeholder {
color: #3a3a3a;
}

.login-form-field {
border: none;
border-bottom: 1px solid #3a3a3a;
```

```

margin-bottom: 10px;
border-radius: 3px;
outline: none;
padding: 0px 0px 5px 5px;
}

#login-form-submit {
    width: 100%;
    padding: 7px;
    border: none;
    border-radius: 5px;
    color: white;
    font-weight: bold;
    background-color: #3a3a3a;
    cursor: pointer;
    outline: none;
}

```

2. Halaman Must To Buy

- HTML

```

<!DOCTYPE html>
<html>
<head>
    <title>To Do List</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
<div id="myDIV" class="header">
    <h2 class="headertext">Must To Buy</h2>
    <input type="text" id="myInput" placeholder="Type here...">
    <span onclick="newElement()" class="addBtn">Add</span>
</div>

<ul id="myUL">
    <li>Indomie goreng</li>
    <li class="checked">Telur</li>
    <li>Sunlight</li>
    <li>Oreo</li>
    <li>Ice Cream</li>
    <li>Rinso</li>
    <li class="checked">Bayam</li>
    <li>Astor</li>
</ul>

<script>
// Menambahkan tombol "x" pada setiap list
var myNodelist = document.getElementsByTagName("LI");
var i;
for (i = 0; i < myNodelist.length; i++) {
    var span = document.createElement("SPAN");

```

```

var txt = document.createTextNode("\u00D7");
span.className = "close";
span.appendChild(txt);
myNodelist[i].appendChild(span);
}

// Menghapus list saat tombol "x" diklik
var close = document.getElementsByClassName("close");
var i;
for (i = 0; i < close.length; i++) {
    close[i].onclick = function() {
        var div = this.parentElement;
        div.style.display = "none";
    }
}

// Menambahkan "checklist" pada list yang dipilih
var list = document.querySelector('ul');
list.addEventListener('click', function(ev) {
    if (ev.target.tagName === 'LI') {
        ev.target.classList.toggle('checked');
    }
}, false);

// Menambahkan list baru dengan tombol "add"
function newElement() {
    var li = document.createElement("li");
    var inputValue = document.getElementById("myInput").value;
    var t = document.createTextNode(inputValue);
    li.appendChild(t);
    if (inputValue === '') {
        alert("You must write something!");
    } else {
        document.getElementById("myUL").appendChild(li);
    }
    document.getElementById("myInput").value = "";
}

// Menambahkan tombol "x" pada list yang baru ditambahkan
var span = document.createElement("SPAN");
var txt = document.createTextNode("\u00D7");
span.className = "close";
span.appendChild(txt);
li.appendChild(span);

// Menghapus list yang baru dibuat saat tombol "x" diklik
for (i = 0; i < close.length; i++) {
    close[i].onclick = function() {
        var div = this.parentElement;
        div.style.display = "none";
    }
}
}

</script>

</body>
</html>

```

- CSS

```
body {
    margin: 0;
    min-width: 250px;
    font-family: Arial, Helvetica, sans-serif;
}

/* Include the padding and border in an element's total width
and height */
* {
    box-sizing: border-box;
}

/* Remove margins and padding from the list */
ul {
    margin: 0;
    padding: 0;
}

/* Style the list items */
ul li {
    cursor: pointer;
    position: relative;
    padding: 12px 8px 12px 40px;
    list-style-type: none;
    background: #eee;
    font-size: 18px;
    transition: 0.2s;

    /* make the list items unselectable */
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
}

/* Set all odd list items to a different color (zebra-stripes)
*/
ul li:nth-child(odd) {
    background: #f9f9f9;
}

/* Darker background-color on hover */
ul li:hover {
    background: #ddd;
}

/* When clicked on, add a background color and strike out text
*/
ul li.checked {
    background: #888;
    color: #fff;
    text-decoration: line-through;
}

/* Add a "checked" mark when clicked on */

```

```
ul li.checked::before {
    content: '';
    position: absolute;
    border-color: #fff;
    border-style: solid;
    border-width: 0 2px 2px 0;
    top: 10px;
    left: 16px;
    transform: rotate(45deg);
    height: 15px;
    width: 7px;
}

/* Style the close button */
.close {
    position: absolute;
    right: 0;
    top: 0;
    padding: 12px 16px 12px 16px;
}

.close:hover {
    background-color: #8c0e0e;
    color: white;
}

/* Style the header */
.header {
    background-image: url(groceries.jfif);
    padding: 40px 40px;
    color: rgb(0, 0, 0);
    text-align: center;
}

.headertext{
    margin:15px;
    background-color:hsl(0, 0%, 100%);
    width: 200px;
    padding: 10px;
    margin: auto;
    margin-bottom: 25px;

}

/* Clear floats after the header */
.header:after {
    content: "";
    display: table;
    clear: both;
}

/* Style the input */
input {
    margin: 0;
    border: none;
    border-radius: 0;
    width: 75%;
```

```
padding: 10px;
float: left;
font-size: 16px;
}

/* Style the "Add" button */
.addButton {
    padding: 10px;
    width: 25%;
    background: #d9d9d9;
    color: #555;
    float: left;
    text-align: center;
    font-size: 16px;
    cursor: pointer;
    transition: 0.3s;
    border-radius: 0;
}

.addButton:hover {
    background-color: #bbb;
}
```