

Quiz1

(Login Form dan Menu Form)



Disusun Oleh :

Arief Nauvan Ramadha
Junior Web Developer A

**PELATIHAN VSGA
JUNIOR WEB DEVELOPER
DIGITAL TALENT SCHOLARSHIP
2021**

A. Membuat Login Form

* Source Code

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <link rel="stylesheet" type="text/css" href="css.css">
    <link rel="stylesheet" type="text/css" href="js.js">
</head>
<body>
    <div class="container">
        <form action="profile.html" method="get" id="myForm"
onsubmit="return validasi()" class="form">
            <div class="error" style="display: none;"></div>
            <h2>LOGIN</h2>
            <div class="input-field" style="margin-right: 25px;">
                <label for="email">Email</label>
                <input type="text" id="email" />
            </div>

            <div class="input-field" style="margin-right: 25px;">
                <label for="password" style="margin-left: 15px; margin-top:
10px;">Password</label>
                <input type="password" id="password" />
            </div>
            <input class="login" type="submit" name="submit" value="submit">
        </form>
        <script type="text/javascript" src="js.js"></script>
    </div>
</body>
</html>
```

CSS

```
body{
    overflow-y: hidden;
    margin:0;
    padding: 0;
    font-family: verdana;
    background-image: linear-gradient(orange, red);
}

form{
    font-family: arial;
    background-color: rgb(255,255,255,0.7);
    width: 300px;
    margin: 200px auto;
```

```
margin-left, margin-right: auto;
padding: 30px;
text-align: right;
color: #2D2D2D;
border-radius: 10px 40px;
box-shadow: 2px 2px 7px #393939;
font-weight: bold;
}

label {
display: block;
margin-bottom: 6px;
margin-right: 215px;
font-weight: bold;
}

input{
margin: 10px;
border-radius: 2.5px;
}

input.login{
background-image: linear-gradient(45deg,orange, red);
color: white;
font-family: verdana;
font-weight: bold;
border-radius: 10px;
padding: 5px 10px;
margin: 30px 10px auto 0px;
text-transform: uppercase;
transition: .2s;
}

input.login:hover{
background-color: #93A4B0;
color: #CDCDCD;
}

input[type="text"]{
width: 100%;
border-radius: 4px;
border: 1px solid #999;
padding: 6px;
margin-bottom: 12px;
outline: none;
}

input[type="password"] {
width: 100%;
border-radius: 4px;
border: 1px solid #999;
```

```

padding: 6px;
margin-bottom: 12px;
outline: none;
}

h2{
    text-align:center;
    margin: 50px auto;
    margin-top: 10px;
    padding: 10px;
    font-weight: bold;
    color: white;
    font-family: verdana;
    background-color: orangered;
    border-radius: 20px;
}

.error {
    border: 15px solid #ff6c6c;
    border-radius: 10px;
    padding: 15px;
    margin-bottom: 18px;
    background: #ff7272;
    color: #fff;
}

```

Javascript

```

var $form = document.querySelector('.form');
var $error = document.querySelector('.error');
var $inputFields = document.querySelectorAll('.input-field');
var $email = document.getElementById('email');
var $password = document.getElementById('password');

function addError(message) {
    $error.innerHTML = message;
    $error.style.display = 'block';
}

function removeError() {
    $error.innerHTML = "";
    $error.style.display = 'hidden';
}

function validate(event) {
    // event.preventDefault() untuk tidak menjalankan fungsi form apabila di submit,
    yaitu mengirim data ke 'action'.
    event.preventDefault();
    $error.style.display = 'none';
}

```

```

        if ($email.value !== 'coba123@gmail.com' || $password.value !==
'CobaLogin123') {
            addError("Email atau password salah");
        }else {
            removeError();
            alert("Anda Berhasil Login!");
        }
    }
}

$form.addEventListener('submit', validate);

```

* screenshot source code login.html

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Login</title>
5      <link rel="stylesheet" type="text/css" href="css.css">
6      <link rel="stylesheet" type="text/css" href="js.js">
7  </head>
8  <body>
9      <div class="container">
10         <form action="profile.html" method="get" id="myForm" onsubmit="return validasi()" class="
form">
11             <div class="error" style="display: none;"></div>
12             <h2>LOGIN</h2>
13             <div class="input-field" style="margin-right: 25px;">
14                 <label for="email">Email</label>
15                 <input type="text" id="email" />
16             </div>
17             <div class="input-field" style="margin-right: 25px;">
18                 <label for="password" style="margin-left: 15px; margin-top: 10px;">Password</label>
19                 <input type="password" id="password" />
20             </div>
21             <input class="login" type="submit" name="submit" value="submit">
22         </form>
23         <script type="text/javascript" src="js.js"></script>
24     </div>
25 </body>
26 </html>

```

* Penjelasan

1. Pada tag head terdapat <link> yang mengarah kepada file css.css agar tampilan dari html sesuai yang diinginkan.
2. Pada tag body terdapat class container, yang digunakan untuk membuat login form.
3. Didalam container dibuat sebuah class bernama error sebagai tempat untuk memasukkan pesan error pada saat login gagal. Class tersebut dibuat dengan display none atau sengaja disembunyikan, dan baru akan ditampilkan jika terjadi error yang dimaksud.
4. Di dalam container dibuat class Bernama input-form, yang digunakan sebagai untuk memasukkan email dan password
5. Di dalam input-form dibuat tag label dan input. Label berfungsi untuk memberi nama input form, sedangkan input digunakan untuk memasukkan sebuah nilai kedalam id (email dan password).
6. Yang terakhir dibuatlah tag script yang mengarah ke app.js. Tag ini berfungsi untuk menghubungkan login.html dengan proses validasi yang akan dilakukan oleh file javascript.

* screenshot source code css.css

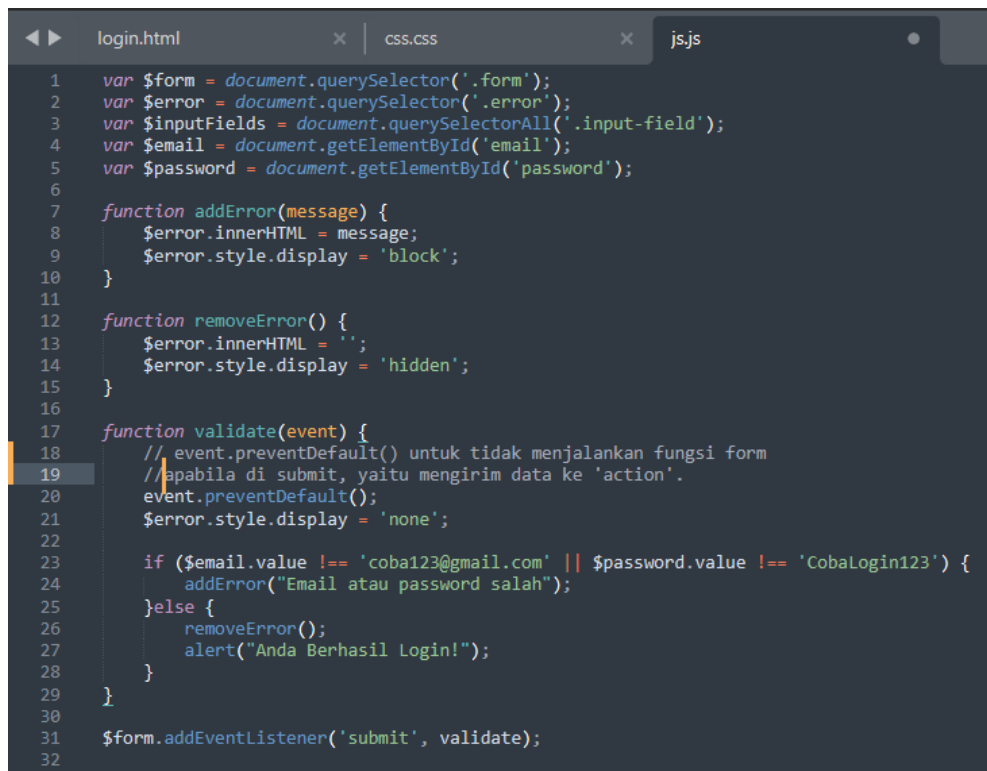
```
login.html x css.css x js.js x
1  body{
2      overflow-y: hidden;
3      margin:0;
4      padding: 0;
5      font-family: verdana;
6      background-image: linear-gradient(orange, red);
7  }
8
9  form{
10     font-family: arial;
11     background-color: rgb(255,255,255,0.7);
12     width: 300px;
13     margin: 200px auto;
14     margin-left, margin-right: auto;
15     padding: 30px;
16     text-align: right;
17     color: #2D2D2D;
18     border-radius: 10px 40px;
19     box-shadow: 2px 2px 7px #393939;
20     font-weight: bold;
21 }
22
23 label {
24     display: block;
25     margin-bottom: 6px;
26     margin-right: 215px;
27     font-weight: bold;
28 }
29
30 input{
31     margin: 10px;
32     border-radius: 2.5px;
33 }
34
35 input.login{
36     background-image: linear-gradient(45deg,orange, red);
37     color: white;
38     font-family: verdana;
39     font-weight: bold;
40     border-radius: 10px;
41     padding: 5px 10px;
42     margin: 30px 10px auto 0px;
43     text-transform: uppercase;
44     transition: .2s;
45 }
46
47 input.login:hover{
48     background-color: #93A4B0;
49     color: #CDCDCD;
50 }
51
52 input[type="text"]{
53     width: 100%;
54     border-radius: 4px;
55     border: 1px solid #999;
```

```
login.html x CSS.CSS x js.js
55     border: 1px solid #999;
56     padding: 6px;
57     margin-bottom: 12px;
58     outline: none;
59 }
60
61 input[type="password"] {
62     width: 100%;
63     border-radius: 4px;
64     border: 1px solid #999;
65     padding: 6px;
66     margin-bottom: 12px;
67     outline: none;
68 }
69
70 h2{
71     text-align:center;
72     margin: 50px auto;
73     margin-top: 10px;
74     padding: 10px;
75     font-weight: bold;
76     color: white;
77     font-family: verdana;
78     background-color: orangered;
79     border-radius: 20px;
80 }
81
82 .error {
83     border: 15px solid #ff6c6c;
84     border-radius: 10px;
85     padding: 15px;
86     margin-bottom: 18px;
87     background: #ff7272;
88     color: #fff;
89 }
90 }
```

* Penjelasan

1. Bagian body disetting dengan font family verdana, dengan overflow-y hidden, margin 0, padding 0, dan background-image: linear-gradient(orange, red)
2. Pada bagian form disetting font-family arial, background-color rgb(255,255,255,0.7), width 300px, margin 200px auto, margin-left dan margin-right auto, padding 30px, text-align right, color #2D2D2D, border-radius 10 px 40px, box-shadow 2px 2px 7px #393939
3. Bagian Label terdapat display block, margin-bottom 6px, margin-right 215px, dan font-weight bold
4. Bagian input terdapat margin 10px, dan border-radius 2.5px
5. Bagian input.login terdapat background-image: linear-gradient(45deg,orange, red); color: white; font-family: verdana; font-weight: bold; border-radius: 10px; padding: 5px 10px; margin: 30px 10px auto 0px; text-transform: uppercase; transition: .2s;
6. Bagian input.login.hover terdapat background-color: #93A4B0; color: #CDCDCD;
7. Bagian input [type="text"] dan [type="password"] terdapat width: 100%; border-radius: 4px; border: 1px solid #999; padding: 6px; margin-bottom: 12px; outline: none;
8. Bagian h2 terdapat text-align:center; margin: 50px auto; margin-top: 10px; padding: 10px; font-weight: bold; color: white; font-family: verdana; background-color: orangered; border-radius: 20px;
9. Dan terakhir pada bagian class error terdapat border: 15px solid #ff6c6c; border-radius: 10px; padding: 15px; margin-bottom: 18px; background: #ff7272; color: #fff;

* screenshot source code js.js



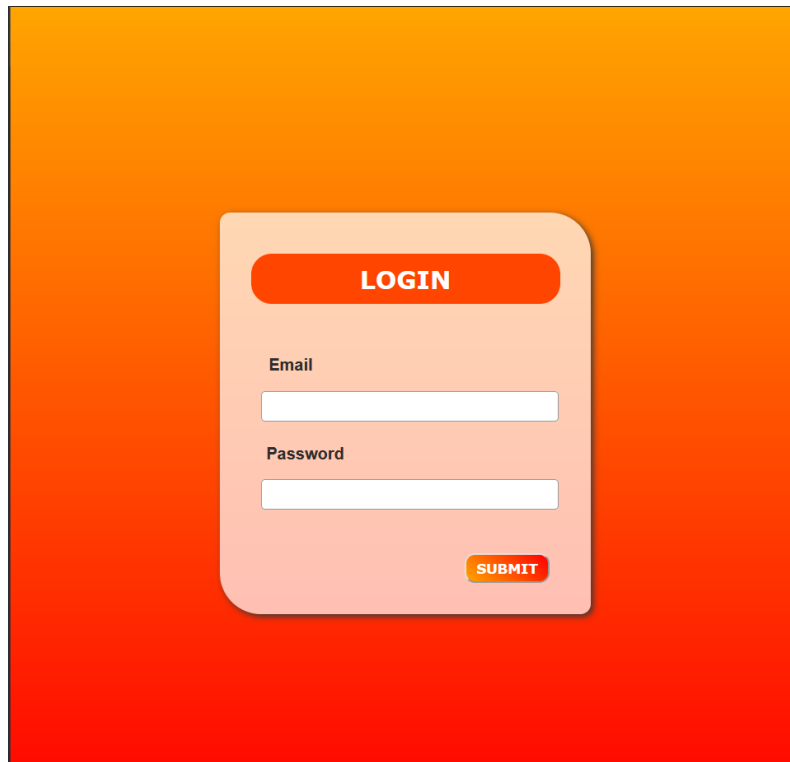
```
1  var $form = document.querySelector('.form');
2  var $error = document.querySelector('.error');
3  var $inputFields = document.querySelectorAll('.input-field');
4  var $email = document.getElementById('email');
5  var $password = document.getElementById('password');
6
7  function addError(message) {
8      $error.innerHTML = message;
9      $error.style.display = 'block';
10 }
11
12 function removeError() {
13     $error.innerHTML = '';
14     $error.style.display = 'hidden';
15 }
16
17 function validate(event) {
18     // event.preventDefault() untuk tidak menjalankan fungsi form
19     // apabila di submit, yaitu mengirim data ke 'action'.
20     event.preventDefault();
21     $error.style.display = 'none';
22
23     if ($email.value !== 'coba123@gmail.com' || $password.value !== 'CobaLogin123') {
24         addError("Email atau password salah");
25     } else {
26         removeError();
27         alert("Anda Berhasil Login!");
28     }
29 }
30
31 $form.addEventListener('submit', validate);
32
```

* Penjelasan

1. Pertama dideklarasikan variable \$form dengan perintah querySelector berfungsi untuk memasukkan form yang telah dibuat tadi agar dapat dikenali oleh javascript.
2. Dideklarasikan variable \$error dengan perintah querySelector berfungsi untuk memasukkan bagian error yang telah dibuat tadi agar dapat dikenali oleh javascript.
3. Dideklarasikan variable \$input-form dengan perintah querySelectorAll berfungsi untuk memasukkan semua class yang dipilih secara sekaligus.
4. Kemudian dideklarasikan variable \$email dan \$password dengan perintah getElement ById yaitu untuk mengambil value dari id tersebut kemudian dimasukkan kedalam variable javascript yang telah dibuat.
5. Selanjutnya dibuat fungsi addError yaitu menambahkan pesan error. Didalamnya terdapat parameter berupa message yaitu pesan yang akan dimasukkan nantinya kedalam variable \$error dengan menggunakan perintah innerHTML. Kemudian diberi style yaitu dengan style block.
6. Fungsi yang kedua yaitu removeError atau fungsi yang akan digunakan untuk menghapus pesan error apabila login berhasil dilakukan dengan benar. Caranya yaitu dengan merubah value dari \$error dengan kosong atau ''. Kemudian untuk style nya dibuat hidden.
7. Fungsi yang terakhir yaitu validate atau memvalidasi email dan password yang dimasukkan. Terdapat parameter event yaitu yang terjadi pada saat sebuah javascript dipanggil. Kemudian dibuat event.preventDefault() yaitu berfungsi untuk mencegah dilakukannya proses action pada saat sebuah form disubmit.
8. Masih di dalam validate, error pada mulanya disetting dengan style none, baru kemudian dibuat sebuah kondisi yaitu apabila email dan password tidak sesuai maka yang akan terjadi adalah pemanggilan fungsi addError dengan diberi pesan kesalahan yaitu email atau password salah.

Kemudian apabila email dan password sesuai, maka akan dilakukan fungsi `removeError`, kemudian dipanggil perintah `alert` untuk menampilkan pesan login berhasil

*** Screenshot tampilan Login Form**



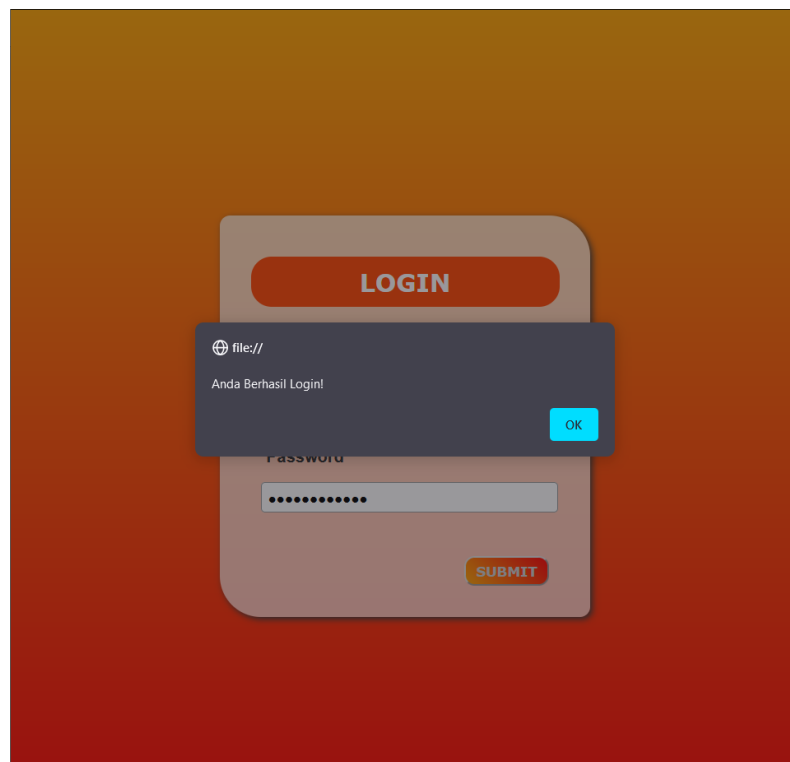
LOGIN

Email

Password

SUBMIT

*** Screenshot tampilan Login Form Ketika Berhasil Login**



LOGIN

file://

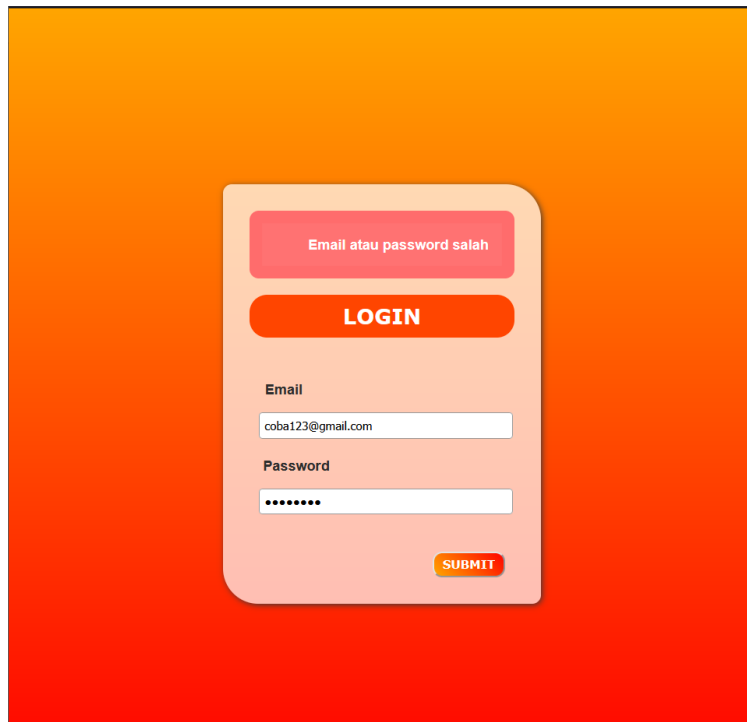
Anda Berhasil Login!

OK

Password

SUBMIT

* Screenshot tampilan Login Form Ketika Email atau Password salah



The screenshot shows a login form centered on a background with a vertical orange-to-red gradient. The form itself is a light orange rounded rectangle. At the top of the form is a pink error message box containing the text "Email atau password salah". Below this is a prominent orange "LOGIN" button. Underneath the button are two input fields: one for "Email" containing the text "coba123@gmail.com" and one for "Password" filled with dots. At the bottom right of the form is a small yellow "SUBMIT" button.

Email atau password salah

LOGIN

Email
coba123@gmail.com

Password
••••••••

SUBMIT

B. Membuat Menu Form

* Source Code

HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Menu</title>
  <link rel="stylesheet" type="text/css" href="css.css">
</head>
<body>
  <div class="app"></div>
  <script type="text/javascript" src="js.js"></script>
</body>
</html>
```

CSS

```
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}

body {
  font-family: sans-serif;
  text-align: center;
  color: #333;
  padding-bottom: 40px;
  background-image: linear-gradient(orange, red);
}

a,
a:visited {
  outline: none;
}

a:hover {
  text-decoration: none;
}

.app {
  position: relative;
  min-height: 500px;
  margin: 100px auto;
  margin-left: auto;
  margin-right: auto;
```

```
padding: 32px;
background: lightgray;
border: 1px solid #ddd;
border-radius: 10px 40px;
box-shadow: 2px 2px 7px #393939;
width: 500px;
border-bottom-width: 8px;
border-top-width: 8px;
background: #f5f5f5;
}
```

```
h1 {
margin-bottom: 36px;
font-family: sans-serif;
text-align:center;
margin: 50px auto;
margin-top: 10px;
padding: 10px;
font-weight: bold;
color: white;
background-color: orangered;
border-radius: 15px 5px;
}
```

```
ul {
width: 350px;
margin: 0 auto;
list-style: none;
}
```

```
ul li {
margin-bottom: 8px;
border: 1px solid #ddd;
padding: 12px 16px;
background: #fff;
font-weight: bold;
text-align: left;
}
```

```
ul li.is-active {
color: #fff;
background: #2ECC40;
border-color: #27b137;
}
```

```
ul li.is-active:hover {
background: #69d845;
}
```

```
ul li:hover {
background: #f5f5f5;
```

```

    cursor: pointer;
}

ul li span {
    float: right;
}

.total {
    margin: 0 auto;
    width: 350px;
    padding: 20px 16px;
    font-weight: bold;
    font-size: 1.2em;
    text-align: left;
}

.total span {
    float: right;
}

```

JavaScript

```

function hasClass(el, className) {
    if (el.classList)
        return el.classList.contains(className);
    return !!el.className.match(new RegExp('(\\s|^)' + className + '(\\s|$)'));
}

function addClass(el, className) {
    if (el.classList)
        return el.classList.add(className);
    else if (!hasClass(el, className))
        el.className += ' ' + className;
}

function removeClass(el, className) {
    if (el.classList)
        el.classList.remove(className);
    else if (hasClass(el, className)) {
        var reg = new RegExp('(\\s|^)' + className + '(\\s|$)');
        el.className = el.className.replace(reg, ' ');
    }
}

function currency(amount) {
    return 'Rp ' + amount;
}

var products = [
    {
        id: 1,

```

```
    name: 'Cappucino',
    price: 35000,
    active: true,
  },
  {
    id: 2,
    name: 'Green Tea Latte',
    price: 40000,
    active: true
  },
  {
    id: 3,
    name: 'Fish and Chips',
    price: 50000,
    active: false,
  },
  {
    id: 4,
    name: 'Tuna Sandwich',
    price: 45000,
    active: false,
  },
  {
    id: 5,
    name: 'Mineral Water',
    price: 8000,
    active: false,
  },
  {
    id: 6,
    name: 'French Fries',
    price: 18000,
    active: false,
  },
];

var total = 0;
var $app = document.querySelector('.app');

function renderTitle(container) {
  var $title = document.createElement('h1');
  $title.innerHTML = 'Pesanan';
  container.appendChild($title);
}

function addTotal(product, total, isAdd) {
  if (isAdd) {
    total += product.price;
  } else {
    total -= product.price;
  }
}
```

```

    return total;
}

function renderList(container, products) {
    var $orderList = document.createElement('ul');

    products.forEach(function(product) {
        var $product = document.createElement('li');
        var $productPrice = document.createElement('span');

        $productPrice.innerHTML = currency(product.price);
        $product.innerHTML = product.name;
        $product.appendChild($productPrice);

        $orderList.appendChild($product);

        $product.addEventListener('click', function(event) {

            var isAdd = !hasClass($product, 'is-active');

            if (isAdd) {
                addClass($product, 'is-active');
            } else {
                removeClass($product, 'is-active');
            }

            total = addTotal(product, total, isAdd);

            var $total = document.querySelector('.total span');
            $total.innerHTML = currency(total);
        });
    });

    container.appendChild($orderList);
}

function renderTotalContainer(container, total) {
    var $totalContainer = document.createElement('div');
    addClass($totalContainer, 'total');

    $totalContainer.innerHTML = 'Total: ';

    var $total = document.createElement('span');
    $total.innerHTML = currency(total);
    $totalContainer.appendChild($total);

    container.appendChild($totalContainer);
}

renderTitle($app);
renderList($app, products);

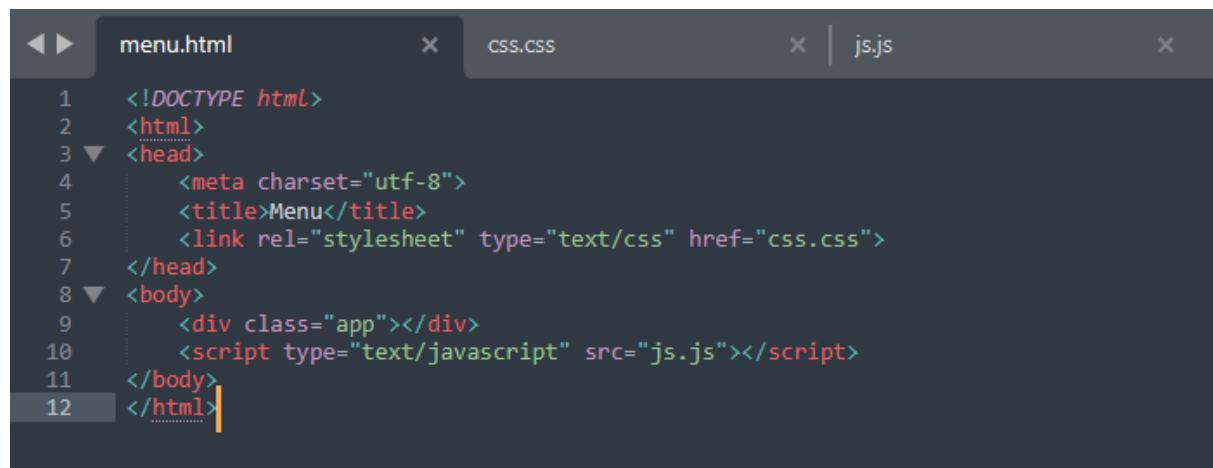
```

```
renderTotalContainer($app, total);

var $products = document.querySelectorAll('li');
$products.forEach(function($product, index) {

    if (index < 2) {
        $product.dispatchEvent(new Event('click'));
    }
});
```

*** screenshot source code menu.html**



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Menu</title>
6      <link rel="stylesheet" type="text/css" href="css.css">
7  </head>
8  <body>
9      <div class="app"></div>
10     <script type="text/javascript" src="js.js"></script>
11 </body>
12 </html>
```

*** Penjelasan**

1. Pada tag head didefinisikan mengenai link yang mengarah kepada tampilan.css agar tampilan dari html sesuai yang diinginkan.
2. Terdapat title pada head yang digunakan untuk menunjukkan nama di halaman web
3. Yang terakhir dibuatlah tag script yang mengarah ke app.js. Tag ini berfungsi untuk menghubungkan menu.html dengan proses penghitungan yang akan dilakukan oleh file javascript.

* screenshot source code css.css

```
1  * {
2      padding: 0;
3      margin: 0;
4      box-sizing: border-box;
5  }
6
7  body {
8      font-family: sans-serif;
9      text-align: center;
10     color: #333;
11     padding-bottom: 40px;
12     background-image: linear-gradient(orange, red);
13 }
14
15 a,
16 a:visited {
17     outline: none;
18 }
19
20 a:hover {
21     text-decoration: none;
22 }
23
24 .app {
25     position: relative;
26     min-height: 500px;
27     margin: 100px auto;
28     margin-left: auto;
29     margin-right: auto;
30     padding: 32px;
31     background: lightgray;
32     border: 1px solid #ddd;
33     border-radius: 10px 40px;
34     box-shadow: 2px 2px 7px #393939;
35     width: 500px;
36     border-bottom-width: 8px;
37     border-top-width: 8px;
38     background: #f5f5f5;
39 }
40
41 h1 {
42     margin-bottom: 36px;
43     font-family: sans-serif;
44     text-align: center;
45     margin: 50px auto;
46     margin-top: 10px;
47     padding: 10px;
48     font-weight: bold;
49     color: white;
50     background-color: orangered;
51     border-radius: 15px 5px;
52 }
53
54 ul {
```

```
menu.html x CSS.CSS x js.js x
55     width: 350px;
56     margin: 0 auto;
57     list-style: none;
58 }
59
60 ▼ ul li {
61     margin-bottom: 8px;
62     border: 1px solid #ddd;
63     padding: 12px 16px;
64     background: #fff;
65     font-weight: bold;
66     text-align: left;
67 }
68
69 ▼ ul li.is-active {
70     color: #fff;
71     background: #2ECC40;
72     border-color: #27b137;
73 }
74
75     ul li.is-active:hover {
76         background: #69d845;
77     }
78
79 ▼ ul li:hover {
80     background: #f5f5f5;
81     cursor: pointer;
82 }
83
84     ul li span {
85         float: right;
86     }
87
88 ▼ .total {
89     margin: 0 auto;
90     width: 350px;
91     padding: 20px 16px;
92     font-weight: bold;
93     font-size: 1.2em;
94     text-align: left;
95 }
96
97     .total span {
98         float: right;
99 }
```

* Penjelasan

1. * digunakan untuk melakukan styling pada seluruh bagian yaitu dengan margin dan padding auto 0 dan dibuat box sizing border-box.
2. Body digunakan untuk styling pada body sehingga tampilan sesuai dengan soal.
3. .app yaitu digunakan untuk styling pada class app sedemikian rupa hingga sesuai dengan soal.
4. H1 disetting dengan margin bottom supaya ada jarak dengan konten dibawahnya.
5. Ul untuk styling pada unorderedlist.
6. Kemudian ul li untuk masing" listnya. Ketika list di klik atau .is-active maka akan dibuat berwarna hijau, dan ketika dilintasi mouse (hover) akan disetting berwarna abu-abu pada saat non aktif dan hijau agak muda pada saat sudah aktif.

7. UI li span yaitu setingan display terhadap harga, dibuat float right atau rata kanan agar harga yang dituliskan mepet ke kanan.
8. .total span yaitu styling pada harga total juga dibuat float right agar total harga menjadi rata kanan.

* screenshot source code js.js

```
menu.html | css.css | js.js
1  function hasClass(el, className) {
2      if (el.classList)
3          return el.classList.contains(className);
4      return !!el.className.match(new RegExp('(\\s|^)' + className + '(\\s|$)'));
5  }
6
7  function addClass(el, className) {
8      if (el.classList)
9          return el.classList.add(className);
10     else if (!hasClass(el, className))
11         el.className += ' ' + className;
12 }
13
14 function removeClass(el, className) {
15     if (el.classList)
16         el.classList.remove(className);
17     else if (hasClass(el, className)) {
18         var reg = new RegExp('(\\s|^)' + className + '(\\s|$)');
19         el.className = el.className.replace(reg, ' ');
20     }
21 }
22
23 function currency(amount) {
24     return 'Rp ' + amount;
25 }
26
27 var products = [
28     {
29         id: 1,
30         name: 'Cappucino',
31         price: 35000,
32         active: true,
33     },
34     {
35         id: 2,
36         name: 'Green Tea Latte',
37         price: 40000,
38         active: true
39     },
40     {
41         id: 3,
42         name: 'Fish and Chips',
43         price: 50000,
44         active: false,
45     },
46     {
47         id: 4,
48         name: 'Tuna Sandwich',
49         price: 45000,
50         active: false,
51     },
52     {
53         id: 5,
54         name: 'Mineral Water',
55         price: 8000,
```

```
menu.html x | css.css x | js.js x
56     active: false,
57   },
58   {
59     id: 6,
60     name: 'French Fries',
61     price: 18000,
62     active: false,
63   },
64 ];
65
66 var total = 0;
67 var $app = document.querySelector('.app');
68
69 function renderTitle(container) {
70   var $title = document.createElement('h1');
71   $title.innerHTML = 'Pesanan';
72   container.appendChild($title);
73 }
74
75 function addTotal(product, total, isAdd) {
76   if (isAdd) {
77     total += product.price;
78   } else {
79     total -= product.price;
80   }
81   return total;
82 }
83
84 function renderList(container, products) {
85   var $orderList = document.createElement('ul');
86
87   products.forEach(function(product) {
88     var $product = document.createElement('li');
89     var $productPrice = document.createElement('span');
90
91     $productPrice.innerHTML = currency(product.price);
92     $product.innerHTML = product.name;
93     $product.appendChild($productPrice);
94
95     $orderList.appendChild($product);
96
97     $product.addEventListener('click', function(event) {
98
99       var isAdd = !hasClass($product, 'is-active');
100
101       if (isAdd) {
102         addClass($product, 'is-active');
103       } else {
104         removeClass($product, 'is-active');
105       }
106
107       total = addTotal(product, total, isAdd);
108
109       var $total = document.querySelector('.total span');
110       $total.innerHTML = currency(total);
111     });
112   });
113 }
```

```
menu.html x CSS.CSS x js.js x
111     });
112     });
113
114     container.appendChild($orderList);
115 }
116
117 function renderTotalContainer(container, total) {
118     var $totalContainer = document.createElement('div');
119     addClass($totalContainer, 'total');
120
121     $totalContainer.innerHTML = 'Total: ';
122
123     var $total = document.createElement('span');
124     $total.innerHTML = currency(total);
125     $totalContainer.appendChild($total);
126
127     container.appendChild($totalContainer);
128 }
129
130 renderTitle($app);
131 renderList($app, products);
132 renderTotalContainer($app, total);
133
134 var $products = document.querySelectorAll('li');
135 $products.forEach(function($product, index) {
136
137     if (index < 2) {
138         $product.dispatchEvent(new Event('click'));
139     }
140 });
141
```

* Penjelasan

1. Pertama di javascript harus didefinisikan class nya terlebih dahulu yaitu dengan dungdi hasClass. Kemudian ditambahkan pula fungsi addClass untuk menambahkan kelas dan removeClass untuk menghilangkan kelas.
2. Kemudian dibuat fungsi currency untuk menampilkan harga dalam Rp (rupiah).
3. Disefinisikan variable bertipe array Bernama product yang didalamnya terdapat id produk (id), nama produk (name), harga produk (price), dan status produk (active).
4. Dedefinisikan pula variable total dengan nilai 0, dan variable \$app dengan perintah querySelector untuk mengambil class app dari menu.html.
5. Selanjutnya ada fungsi renderTitle untuk membuat bagian yang akan menampilkan judul container yaitu Pesanan.
6. AddTotal digunakan untuk melakukan penjumlahan maupun pengurangan total. Total akan bertambah pada saat menu di klik, dan akan berkurang pada saat diklik lagi (dibatalkan).
7. renderList berfungsi untuk merender atau membuat tampilan semua produk, dimulai dengan mendefinisikan ul (unorderedlist) kemudian membuat looping foreach untuk menampilkan li (list) dengan isian di sebelah kiri yaitu nama produk dan diberi span, kemudian di sebelah kanan adalah harga produk.
8. Masih didalam looping foreach, dibuat sebuah addEventListener dengan masukan klik, agar pada saat di klik 1x produk akan dipilih dan pada saat di klik untuk kedua kali maka produk tersebut akan dilepas.
9. Kemudian produk yang terpilih akan dijumlahkan dan ditampilkan melalui variable \$total.
10. Fungsi renderTotalContainer dibuat untuk menampilkan variable total tadi ke dalam html.

11. Terakhir, semua fungsi render dipanggil agar semua bagian yang telah dibuat tadi dapat tertampil di browser.
12. Jika diinginkan kita bisa menambahkan perulangan for each untuk mentrigger klik sehingga ada menu yang otomatis terpilih.

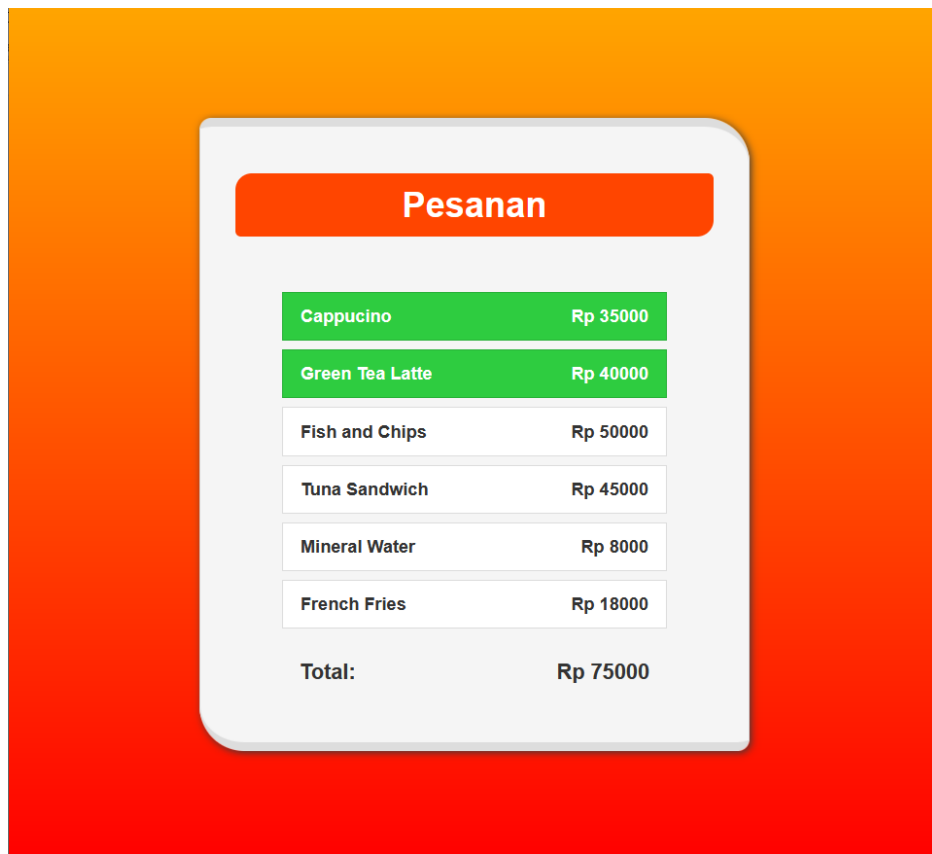
*** Screenshot tampilan Menu Form**



The screenshot shows a web form titled 'Pesanan' (Order) on a red background. The form is a light gray rounded rectangle containing a table of items and their prices. The table has two columns: item names and prices in Indonesian Rupiah (Rp). The items listed are Cappucino, Green Tea Latte, Fish and Chips, Tuna Sandwich, Mineral Water, and French Fries. At the bottom of the table, the total is shown as 'Total: Rp 0'.

Pesanan	
Cappucino	Rp 35000
Green Tea Latte	Rp 40000
Fish and Chips	Rp 50000
Tuna Sandwich	Rp 45000
Mineral Water	Rp 8000
French Fries	Rp 18000
Total:	Rp 0

* Screenshot tampilan saat memilih menu Cappucino dan Green Tea Latte



* Screenshot saat memilih menu Fish and chips dan Tuna Sandwich

