

A. MEMBUAT LOGIN FORM

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title></title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" type="text/css" href="tampilan.css">
  </head>
  <body>
    <div class="container">
      <form action="" class="form">
        <div class="error" style="display: none"></div>

        <div class="input-field">
          <label for="email">Email</label>
          <input type="text" id="email" />
        </div>

        <div class="input-field">
          <label for="password">Password</label>
          <input type="password" id="password" />
        </div>

        <input type="submit" id="submit" value="Login">
      </form>
    </div>
    <script type="text/javascript" src="app.js"></script>
  </body>
</html>
```

Penjelasan :

1. Pada tag head didefinisikan mengenai link yang mengarah kepada tampilan.css agar tampilan dari html sesuai yang diinginkan.
 2. Pada tag body dibuat class container untuk membuat sebuah bagian yang akan dipakai 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 juga dibuat class Bernama input-form yang akan digunakan sebagai tempat 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.

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();
    $error.style.display = 'none';

    if ($email.value !== 'atmayanti@gmail.com' || $password.value !== '12345') {
        addError('Email atau password salah');
    } else {
        removeError();
        alert('Anda Berhasil Login!');
    }
}

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

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.

CSS

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

body {
    font-family: sans-serif;
}

.container {
    margin: 20px auto;
    padding: 32px 44px;
    margin-bottom: 30px;
    background: #f5f5f5;
    border: 1px solid #ddd;
    border-radius: 3px;
    box-shadow: 1px 1px 1px #ddd;
    width: 300px;
    background: #f5f5f5;
}

label {
    display: block;
    margin-bottom: 6px;
}
```

```
#email, #password{
    width: 100%;
    border-radius: 4px;
    border: 1px solid #999;
    padding: 6px;
    margin-bottom: 12px;
    outline: none;
}

#submit {
    width: 100%;
    padding: 6px;
    font-size: 1em;
    background: #0092d8;
    border: 1px solid #0092d8;
    border-radius: 4px;
    outline: none;
    color: #fff;
}

#submit:hover {
    cursor: pointer;
    background: #0376ad;
}

.error {
    border: 1px solid #ff6c6c;
    border-radius: 4px;
    padding: 12px;
    margin-bottom: 18px;
    background: #ff7272;
    color: #fff;
}
```

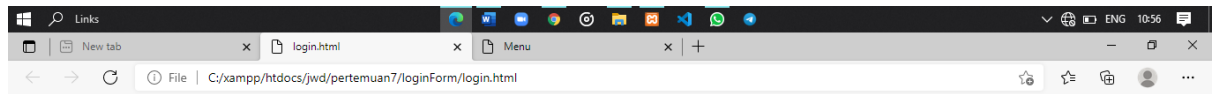
Penjelasan :

1. Bagian * maksudnya adalah all atau semua komponen yang ada di dalam html akan di select dan diberi styling, yaitu dengan padding dan margin 0 dan box-sizing border-box.
 2. Bagian body disetting dengan font family sans-serif
 3. Class container disetting sedemikian rupa hingga sesuai dengan soal.
 4. Label disetting dengan display block supaya otomatis terdapat enter setelahnya. Kemudian diberi margin bottom supaya tidak terlalu mepet dengan input form.
 5. Id email dan password disetting sedemikian rupa hingga boxnya sesuai dengan soal.
 6. Id submit disetting juga supaya berbentuk seperti yang di soal.
 7. Kemudian submit hover disetting agar pada saat mouse melewati tombol submit, tombol tersebut akan berubah warna.
 8. Yang terakhir adalah setting pada class error yaitu setting untuk tampilan pesan kesalahan yang akan ditampilkan diatas input form.
-

Nama : Atmayanti
Kelas : Web A

HASIL :

Tampilan awal :

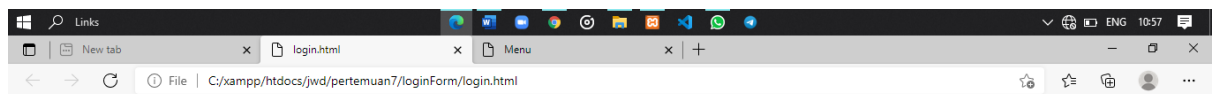


Email

Password

Login

Saat login berhasil :

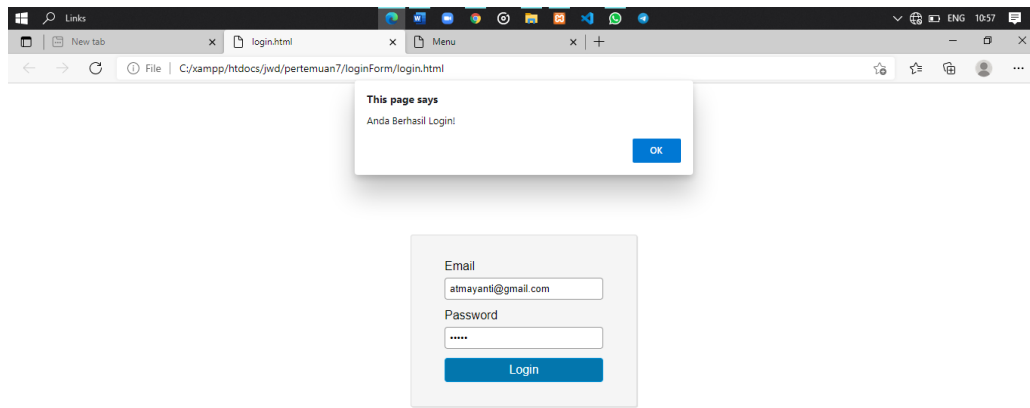


Email

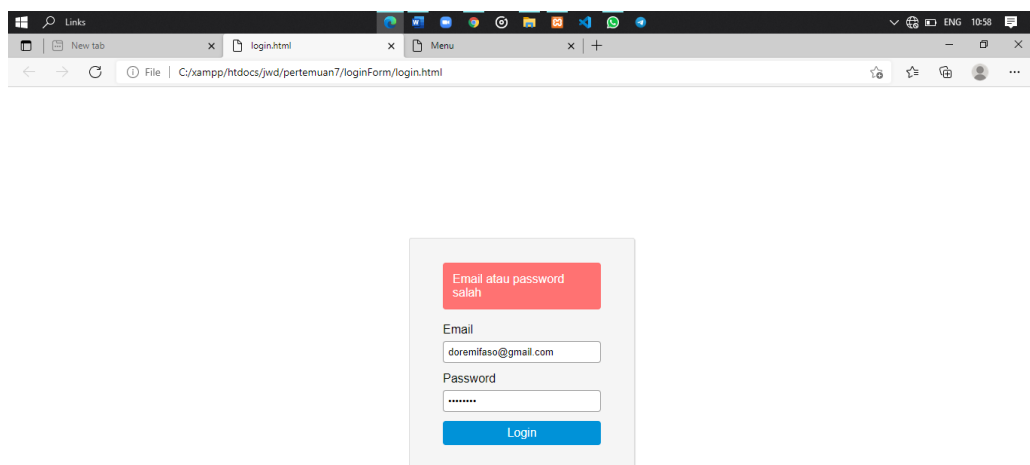
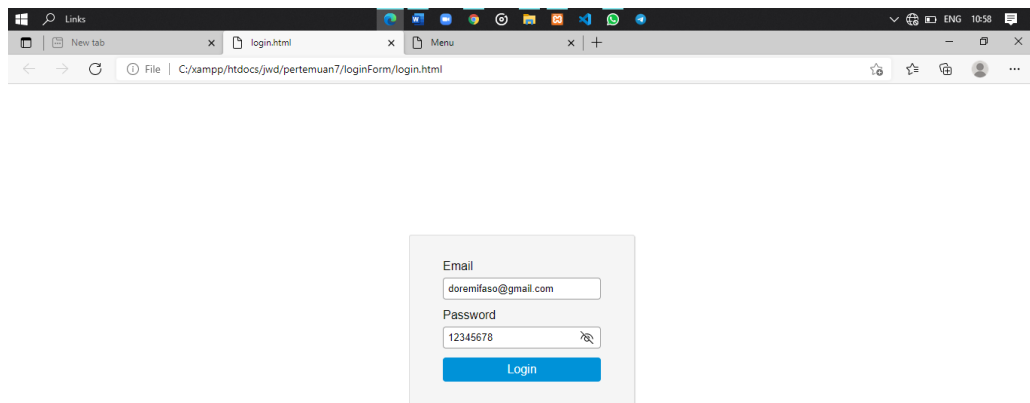
Password

Login

Nama : Atmayanti
Kelas : Web A



Saat login gagal :



B. MEMBUAT ORDER FORM

HTML

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

Penjelasan :

1. Pada tag head didefinisikan mengenai link yang mengarah kepada tampilan.css agar tampilan dari html sesuai yang diinginkan.
2. 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.

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: true,
  },
  {
    id: 4,
    name: 'Tuna Sandwich',
    price: 45000,
    active: true,
  },
  {
    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');

// Render title
```



```
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'));
    }
});
```

Penjelasan :

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

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

CSS

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

body {
  font-family: sans-serif;
  text-align: center;
  color: #333;
  padding-bottom: 40px;
}

a,
a:visited {
  outline: none;
}

a:hover {
  text-decoration: none;
}

.app {
  position: relative;
  min-height: 500px;
  margin: 30px auto;
  padding: 32px;
  background: #f5f5f5;
  border: 1px solid #ddd;
  border-radius: 3px;
  box-shadow: 1px 1px 1px #ddd;
  width: 500px;
  background: #f5f5f5;
}

h1 {
  margin-bottom: 36px;
}

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;
}

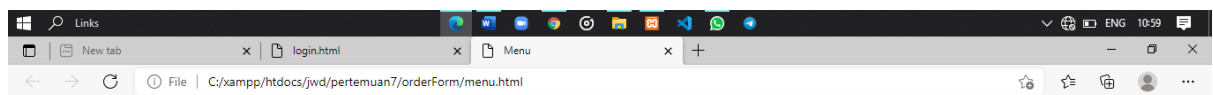
.total span {
  float: right;
}
```

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. Ul 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.

HASIL :

Pada saat tidak ada yang terselect :

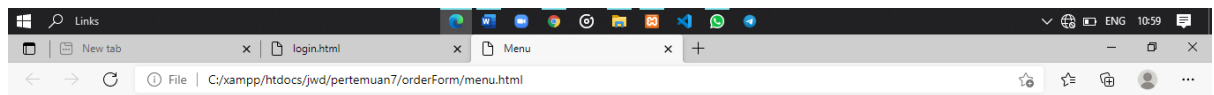


Pesanan	
Cappucino	Rp 35000
Green Tea Latte	Rp 40000
Fish and Chips	Rp 50000
Tuna Sandwich	Rp 45000
Mineral Water	Rp 8000
French Fries	Rp 18000
Total:	Rp 0

Nama : Atmayanti

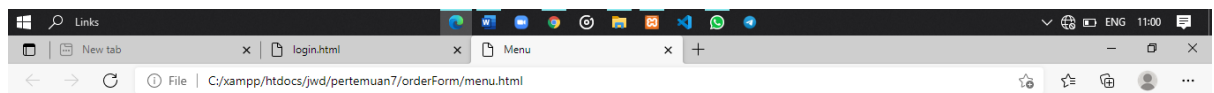
Kelas : Web A

Pada saat terselect 2 menu :



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 90000

Pada saat menu Green Tea dibatalkan :



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 50000