|  |  |  |
| --- | --- | --- |
| Mata Kuliah  Program Studi | **: :** | Pemrograman Web (PW)  D4 – Teknik Informatika **/** D4 – Sistem Informasi Bisnis |
| Semester | **:** | 3 (tiga) |
| Pertemuan ke- | **:** | 13 (tiga belas)  SAFRIZAL RAHMAN  **JOBSHEET 13**  **Web Templating (AdminLTE) dan CRUD** |
| **INFO**    Materi ini adalah materi tambahan, jadi perhatikan dan ikuti tahap demi tahap dari materi ini. | | |

# A. Web Template

Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

Secara umum kita bisa menggunakan template website yang sudah ada di internet untuk mempermudah kita dalam membuat website. Ada banyak sekali template yang ada di internet dan kita ambil satu contoh template untuk halaman administrator, seperti **AdminLTE** template. **AdminLTE** adalah salah satu template yang sering digunakan oleh web developer sebagai template backend/admin pada proyek yang sering dikerjakan. Template ini dibuat menggunakan framework bootsrap yang merupakan framework CSS yang paling banyak digunakan di kalangan web desainer

Template AdminLTE dapat di download di

<https://github.com/ColorlibHQ/AdminLTE/archive/refs/tags/v3.2.0.zip>diakses pada (17 November 2024)

Kita pakai **AdminLTE Versi 3.2.0**

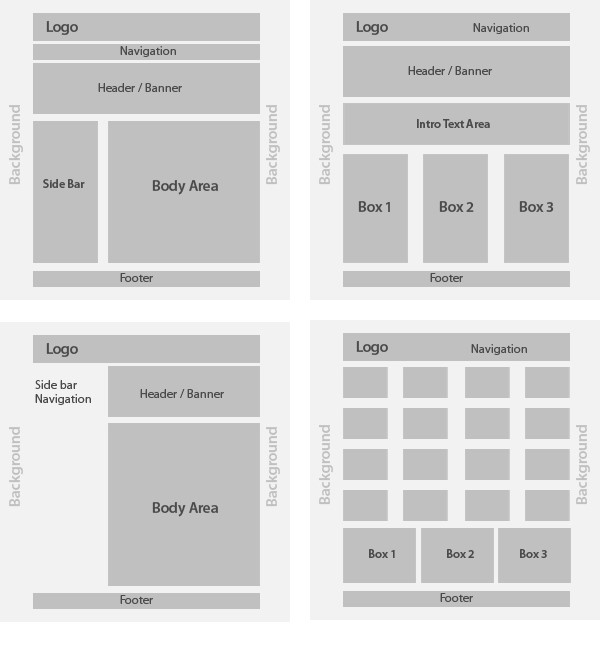
Contoh Web template

|  |  |
| --- | --- |
| **Frontend Template** | **Backend Template** |
|  |  |
|  |  |

# B. Web Layout

Merupakansekumpulan elemen atau variabel desain yang terkait dengan media tertentu untuk mendukung konsep pembuatan website. Pada dasarnya tujuan penggunaan layout website adalah untuk menggabungkan elemen yang digunakan untuk membangun website. Selain elemen di dalamnya, layout website juga harus memuat beberapa elemen, mulai dari header, navigasi, sidebar, hingga footer. Untuk memahami perbedaan elemen sebuah layout website, berikut penjelasan singkatnya

* **Header** → Pada bagian header, layout designer dapat mengisi bagian ini dengan logo website, navigasi situs, ikon media sosial, dan menu pencarian.
* **Navigasi** → Navigasi dapat dipahami sebagai panduan. Di website, navigasi dapat berkisar dari menu yang muncul di bagian atas halaman web hingga menu pendukung lainnya yang biasa ditemukan di bawah situs.
* **Body/Konten** → Selain itu, terdapat elemen body/konten yang biasanya diisi dengan informasi produk, fitur produk, dan deskripsi produk yang dijual.
* **Sidebar** → elemen ini merupakan elemen yang berada di samping kanan/kiri sebuah website. Element ini kadang sudah tidak dipakai, terutama pada Web Frontend. Namun, sidebar masih digunakan di banyak halaman artikel, yang kemudian akan diisi dengan informasi produk, produk terpopuler, dan navigasi tambahan.
* **Footer** → Elemen terakhir adalah footer. Footer merupakan elemen yang berada dibawah sendiri sebagai menu pendukung atau sebagai identitas dari sebuah website.

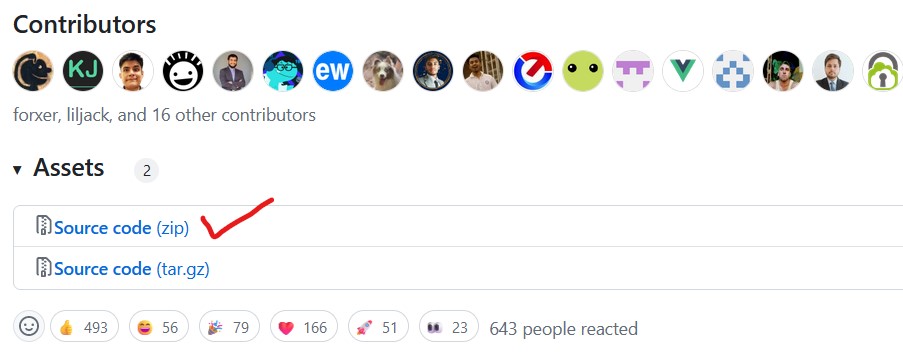
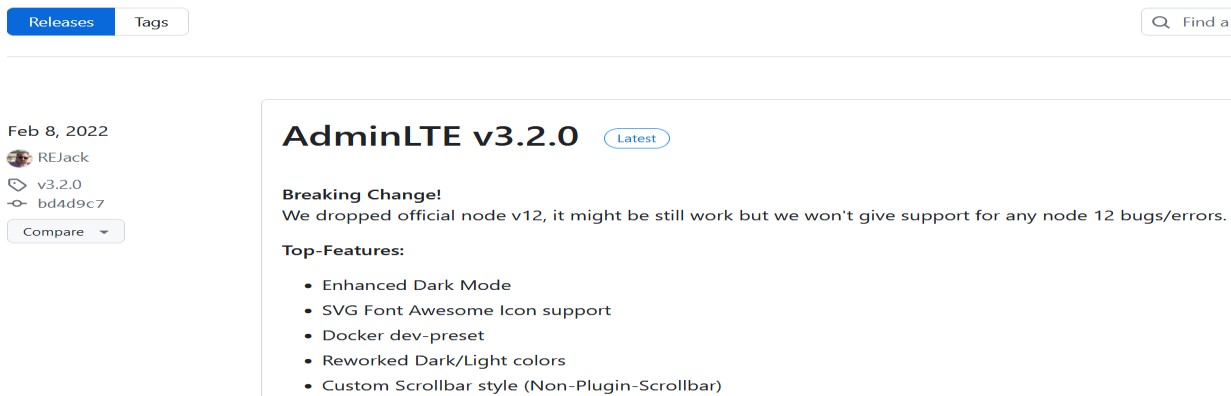


# Layouting AdminLTE

Pada kali ini kita akan melakukan *layouting* template AdminLTE. Tujuan dari *layouting* ini adalah untuk memecah bagian-bagian dari template adminLTE menjadi berbagai elemen. Hal ini dapat mempermudah kita dalam membuat website karena kita bisa menggunakan berulang-ulang elemen yang sudah kita buat nantinya.

**Praktikum 1 - Penerapan AdminLTE (Login):**

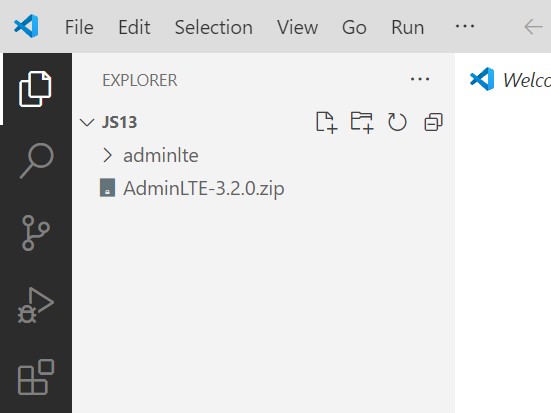
1. Kita download **AdminLTE v3.2.0** yang rilis pada 8 Feb 2022



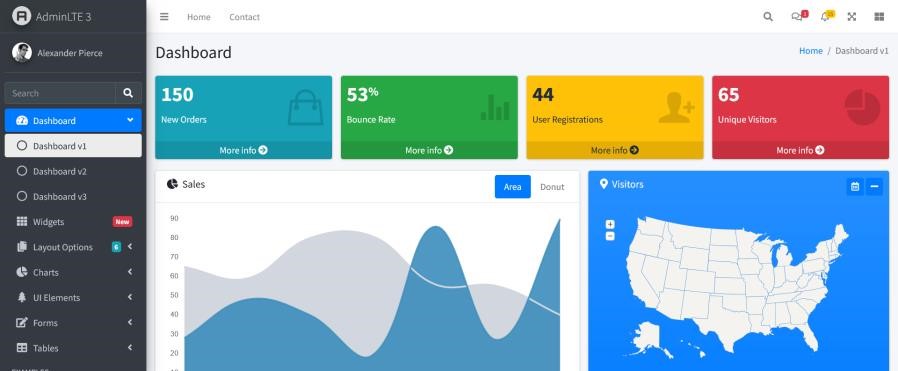
1. Seleteh kita berhasil download, kita ekstrak file yang sudah di download ke folder project C:\laragon\www\dasarWeb\JS13, kemudian kita ***rename*** folder cukup

menjadi

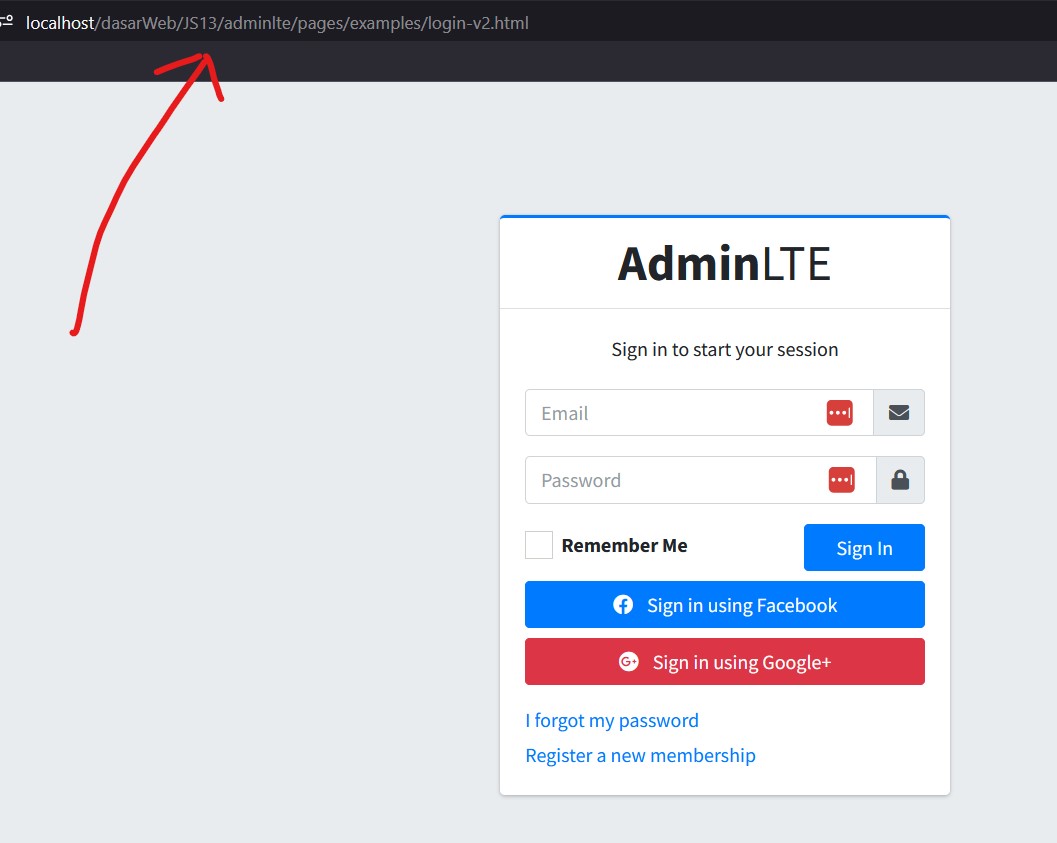
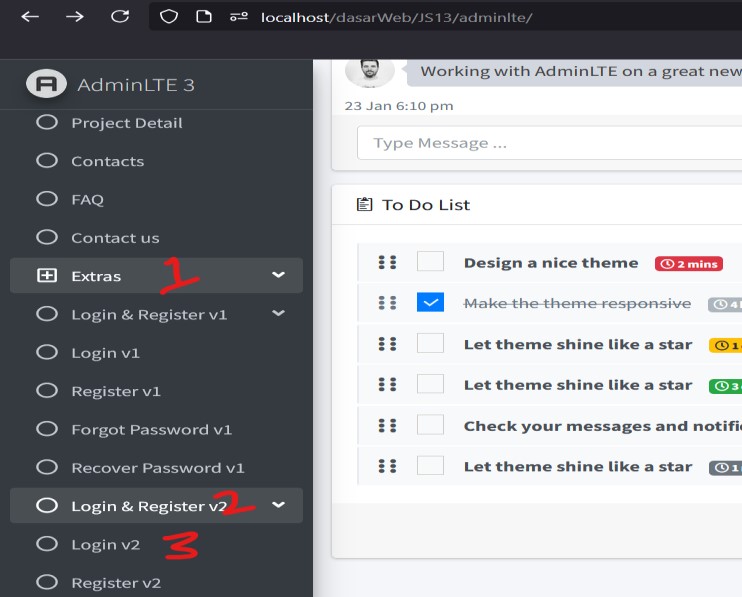
adminlte



1. Kita buat database baru pada phpmyadmin dengan nama dasar\_web. Kemudian kita ***import*** file database dasar\_web.sql pada phpmyadmin.
2. Selanjutnya kita buka di browser dengan alamat <http://localhost/dasarWeb/adminlte>maka akan muncul tampilan seperti berikut



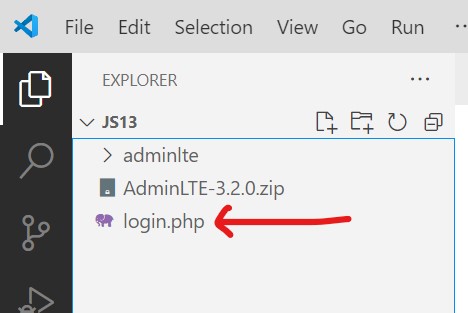
1. Kita klik menu Extras > Login & Register V2 > Login V2, kita gunakan template ini untuk halaman login



1. Kita copy file

C:\laragon\www\dasarWeb\JS13\adminlte\pages\examples\login-v2.html, dan kita paste ke C:\laragon\www\dasarWeb\JS13. Kemudian kita rename menjadi

login.php



1. Kemudian kita edit file login.php tersebut seperti berikut

|  |
| --- |
| <?php  include('lib/Session.php');    $session = new Session();    if($session->get('is\_login') === true){ header('Location: index.php');  }  ?>  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>AdminLTE 3 | Log in (v2)</title>  <!-- Google Font: Source Sans Pro -->  <link rel="stylesheet"  href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallb ack">  <!-- Font Awesome -->  <link rel="stylesheet" href="adminlte/plugins/fontawesome-free/css/all.min.css"> <!-- icheck bootstrap -->  <link rel="stylesheet" href="adminlte/plugins/icheck-bootstrap/icheck-bootstrap.min.css"> <!-- Theme style -->  <link rel="stylesheet" href="adminlte/dist/css/adminlte.min.css"> </head>  <body class="hold-transition login-page">  <div class="login-box">  <!-- /.login-logo -->  <div class="card card-outline card-primary">  <div class="card-header text-center"><a href="/" class="h1"><b>Admin</b>LTE</a></div> <div class="card-body">  <p class="login-box-msg">Sign in to start your session</p>  <?php  $status = $session->getFlash('status');    if($status === false){  $message = $session->getFlash('message'); echo '<div class="alert alert-warning">'. $message . |

|  |
| --- |
| '<button type="button" class="close" data-dismiss="alert" arialabel="Close"><span aria-hidden="true">&times;</span></div>';  }  ?>  <form action="action/auth.php?act=login" method="post" id="form-login">  <div class="input-group mb-3">  <input type="text" class="form-control" name="username" placeholder="Username">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-user"></span>  </div>  </div>  </div>  <div class="input-group mb-3">  <input type="password" class="form-control" name="password" placeholder="Password">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-lock"></span>  </div>  </div>  </div>  <div class="row">  <div class="col-8">  <div class="icheck-primary">  <input type="checkbox" id="remember">  <label for="remember">Remember Me</label>  </div>  </div>  <!-- /.col -->  <div class="col-4">  <button type="submit" class="btn btn-primary btn-block">Sign In</button>  </div>  <!-- /.col -->  </div>  </form>  </div>  <!-- /.card-body -->  </div>  <!-- /.card -->  </div>  <!-- /.login-box -->    <!-- jQuery -->  <script src="adminlte/plugins/jquery/jquery.min.js"></script>  <!-- Bootstrap 4 -->  <script src="adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js"></script> <!-- AdminLTE App -->  <script src="adminlte/dist/js/adminlte.min.js"></script>  <script src="adminlte/plugins/jquery-validation/jquery.validate.min.js"></script>  <script src="adminlte/plugins/jquery-validation/additional-methods.min.js"></script>  <script src="adminlte/plugins/jquery-validation/localization/messages\_id.min.js"></script>    <script>  // maksud nya adl ketika dokumen sudah siap (html telah d render oleh browser) maka jalankan fungsi berikut ini $(document).ready(function(){ $('#form-login').validate({ rules: { username: { required: true, minlength: 3, maxlength: 20  }, password: { required: true, |
| minlength: 5, maxlength: 255  } },  errorElement: 'span',  errorPlacement: function (error, element) { error.addClass('invalid-feedback');  element.closest('.input-group').append(error);  },  highlight: function (element, errorClass, validClass) {  $(element).addClass('is-invalid');  },  unhighlight: function (element, errorClass, validClass) {  $(element).removeClass('is-invalid');  }  });  });  </script>  </body> </html> |

## Pertanyaan 1

Apa saja perbedaan kode program antara kode program pada

C:\laragon\www\dasarWeb\JS13\adminlte\pages\examples\login-v2.html dengan C:\laragon\www\dasarWeb\JS13\login.php ?

Jelaskan pendapatmu

**1. Tipe File**

* **login-v2.html**: Ini adalah file HTML statis. File ini berisi markup HTML yang digunakan untuk menampilkan halaman login. Biasanya, file ini tidak memiliki logika pemrograman di dalamnya dan hanya berfungsi untuk menampilkan konten kepada pengguna.
* **login.php**: Ini adalah file PHP dinamis. File ini dapat berisi kode PHP yang memproses data, berinteraksi dengan database, dan menghasilkan konten HTML secara dinamis berdasarkan input pengguna atau kondisi tertentu.

**2. Fungsi dan Interaksi**

* **login-v2.html**:
  + Hanya menampilkan formulir login dengan elemen-elemen seperti input untuk username dan password, serta tombol untuk login.
  + Tidak memiliki logika untuk memproses data yang dimasukkan oleh pengguna.
  + Jika ada pengalihan atau validasi, itu harus dilakukan dengan JavaScript atau mengarahkan pengguna ke halaman lain.
* **login.php**:
  + Menangani logika pemrosesan login. Ketika pengguna mengisi formulir dan mengirimkannya, file ini akan memproses data tersebut.
  + Dapat melakukan validasi input, memeriksa kredensial pengguna terhadap database, dan mengatur sesi pengguna jika login berhasil.
  + Dapat memberikan umpan balik kepada pengguna jika login gagal (misalnya, pesan kesalahan).

**3. Penggunaan Server**

* **login-v2.html**: Dapat dibuka langsung di browser tanpa memerlukan server. Ini hanya file statis yang tidak memerlukan pemrosesan server.
* **login.php**: Memerlukan server web (seperti Apache atau Nginx) untuk memproses kode PHP. Ketika file ini diakses, server akan menjalankan kode PHP dan mengirimkan hasilnya (HTML) ke browser.

**4. Keamanan**

* **login-v2.html**: Karena tidak ada pemrosesan server, tidak ada langkah-langkah keamanan yang diterapkan di sisi server. Semua validasi harus dilakukan di sisi klien (misalnya, menggunakan JavaScript).
* **login.php**: Dapat menerapkan langkah-langkah keamanan, seperti sanitasi input untuk mencegah serangan SQL injection, serta menggunakan HTTPS untuk melindungi data yang dikirimkan.

**Pendapat**

* **Kelebihan HTML Statis**: File HTML statis seperti **login-v2.html** lebih cepat untuk dimuat dan lebih sederhana untuk dibuat, tetapi tidak memiliki kemampuan untuk memproses data atau berinteraksi dengan database.
* **Kelebihan PHP Dinamis**: File PHP seperti **login.php** lebih fleksibel dan dapat menangani logika bisnis yang lebih kompleks. Ini memungkinkan aplikasi untuk memberikan pengalaman pengguna yang lebih baik dengan memproses data secara real-time dan memberikan umpan balik yang sesuai.
* **Kombinasi Keduanya**: Dalam praktik terbaik, sering kali Anda akan menggunakan file HTML untuk tampilan dan file PHP untuk logika pemrosesan. Anda dapat menggunakan HTML untuk membuat formulir dan kemudian mengirimkan data ke file PHP untuk diproses.

1. Selanjutnya kita buat folder dan file pada

C:\laragon\www\dasarWeb\JS13\lib\Session.php. Kita akan membuat class

Session untuk menangani session PHP. Berikut class Session yang akan kita buat

|  |
| --- |
| <?php class Session{  public function \_\_construct(){  if (session\_status() === PHP\_SESSION\_NONE) { session\_start();  }  }  public function set($key, $value){  $\_SESSION[$key] = $value;  }  public function get($key){  return (isset($\_SESSION[$key]))? $\_SESSION[$key] : null;  }  public function exist($key){  return (isset($\_SESSION[$key]))? true : false;  }  public function delete($key){ if(isset($\_SESSION[$key])){ unset($\_SESSION[$key]); } |
| }  public function setFlash($key, $value){  $\_SESSION['flash'][$key] = $value;  }  public function getFlash($key){  $value = (isset($\_SESSION['flash'][$key]))? $\_SESSION['flash'][$key] : null;  $this->deleteFlash($key); return $value;  }  public function deleteFlash($key){ if(isset($\_SESSION['flash'][$key])){ unset($\_SESSION['flash'][$key]);  }  }  public function deleteAllFlash(){ if(isset($\_SESSION['flash'])){ unset($\_SESSION['flash']);  }  }  public function deleteAll(){ session\_destroy();  }  public function commit(){ session\_commit();  }  } |

1. Kita buat folder dan file di C:\laragon\www\dasarWeb\JS13\lib\Connection.php. Kemudia kita tulis kode pada file Connection.php seperti berikut

|  |
| --- |
| <?php  $username = 'root';  $password = '';  $database = 'dasar\_web';    try{  $db = new mysqli('localhost', $username, $password, $database); if($db->connect\_error){  die('Connection DB failed: ' . $db->connect\_error);  }  }catch(Exception $e){ die($e->getMessage());  }  ?> |

1. Selanjutnya, kita buat kode program untuk menangani proses login dari file login.php yang sudah kita buat. Kita buat folder dan file baru pada

C:\laragon\www\dasarWeb\JS13\action\auth.php. Folder action nanti berfungsi sebagai tempat penyimpanan kode program untuk memproses data. Sekarang kita tulis kode program auth.php seperti berikut

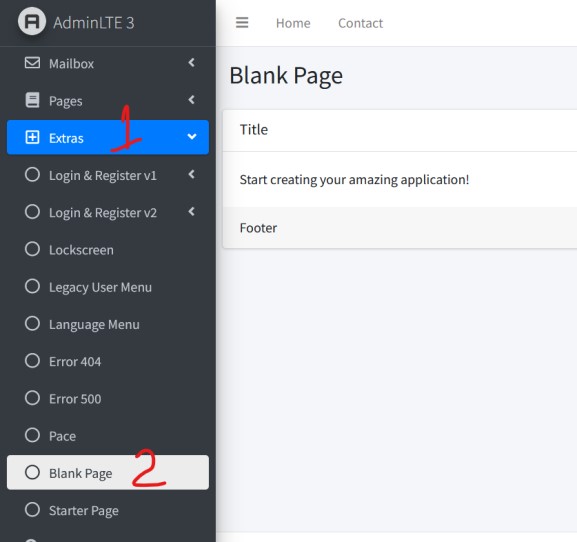
<?php

include('../lib/Session.php'); include('../lib/Connection.php');

|  |
| --- |
| $session = new Session();    $act = isset($\_GET['act'])? strtolower($\_GET['act']) : '';    if($act == 'login'){    $username = $\_POST['username'];  $password = $\_POST['password'];    // digunakan untuk query user  $query = $db->prepare('select \* from m\_user where username = ?');  $query->bind\_param('s', $username);  $query->execute();    // untuk ambil datanya  $data = $query->get\_result()->fetch\_assoc();    // jika password sesuai  if(password\_verify($password, $data['password'])){  $session->set('is\_login', true);  $session->set('username', $data['username']);  $session->set('name', $data['nama']);  $session->set('level', $data['level']);  $session->commit();    header('Location: ../index.php', false);  }else{  $session->setFlash('status', false);  $session->setFlash('message', 'Username dan password salah.');  $session->commit();  header('Location: ../login.php', false);  }  }else if($act == 'logout'){    $session->deleteAll();    header('Location: ../login.php', false); } |

1. Selanjutnya kita lihat pada template adminlte kita di localhost/dasarWeb/JS13/adminlte. Kita pilih menu Extras > Blank Page. Page

inilah yang akan menjadi dasar web template

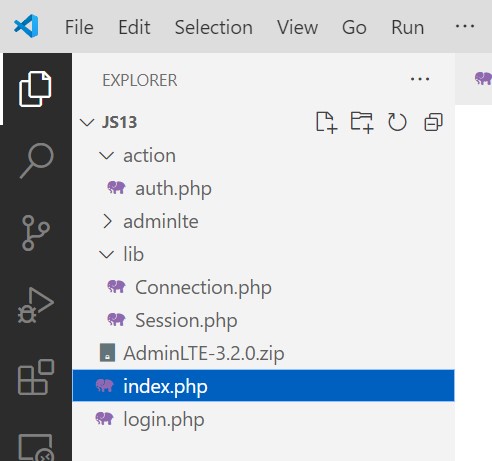


1. Kita *copy* file yang ada di

C:\laragon\www\dasarWeb\JS13\adminlte\pages\examples\blank.html

sesuai dengan alamat url localhost/dasarWeb/JS13/adminlte/pages/examples/blank.html.

Kemudian kita *paste* di C:\laragon\www\dasarWeb\JS13. Selanjutnya kita *rename* file tersebut jadi index.php



1. Selanjutnya kita modifikasi sedikit untuk file index.php yang sudah kita buat. Kita tambahkan kode berikut di awal file index.php

<?php

include('lib/Session.php');

$session = new Session(); if($session->get('is\_login') !== true){ header('Location: login.php');

}

?>

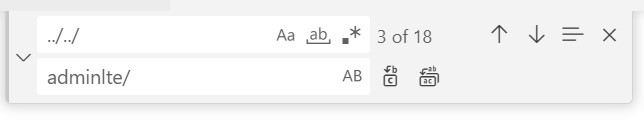
Kemudian kita ***find*** and ***replace*** bagian berikut pada file index.php di VS Code

*Find*

../../

Dan kita *replace* menjadi

adminlte/

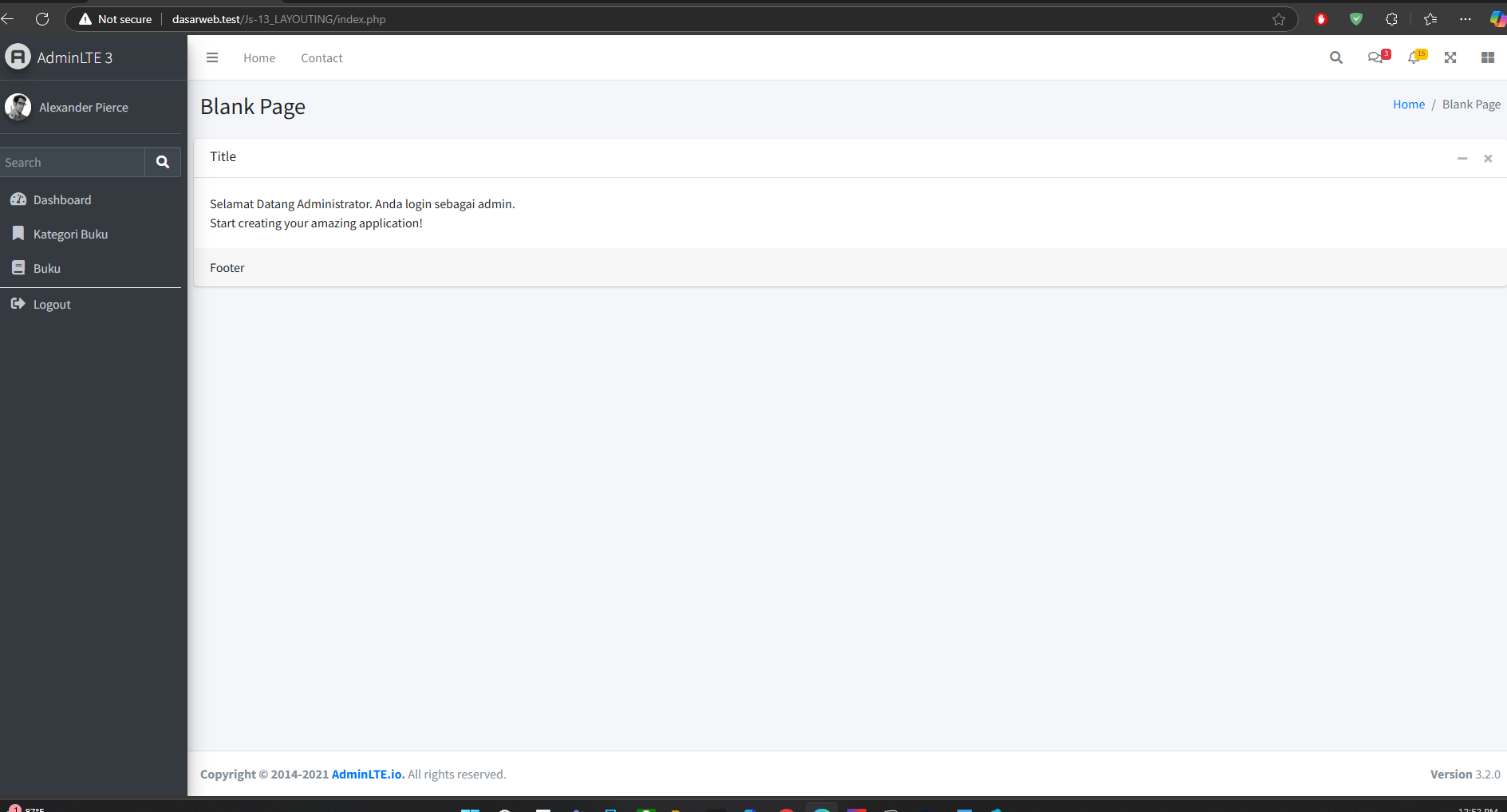


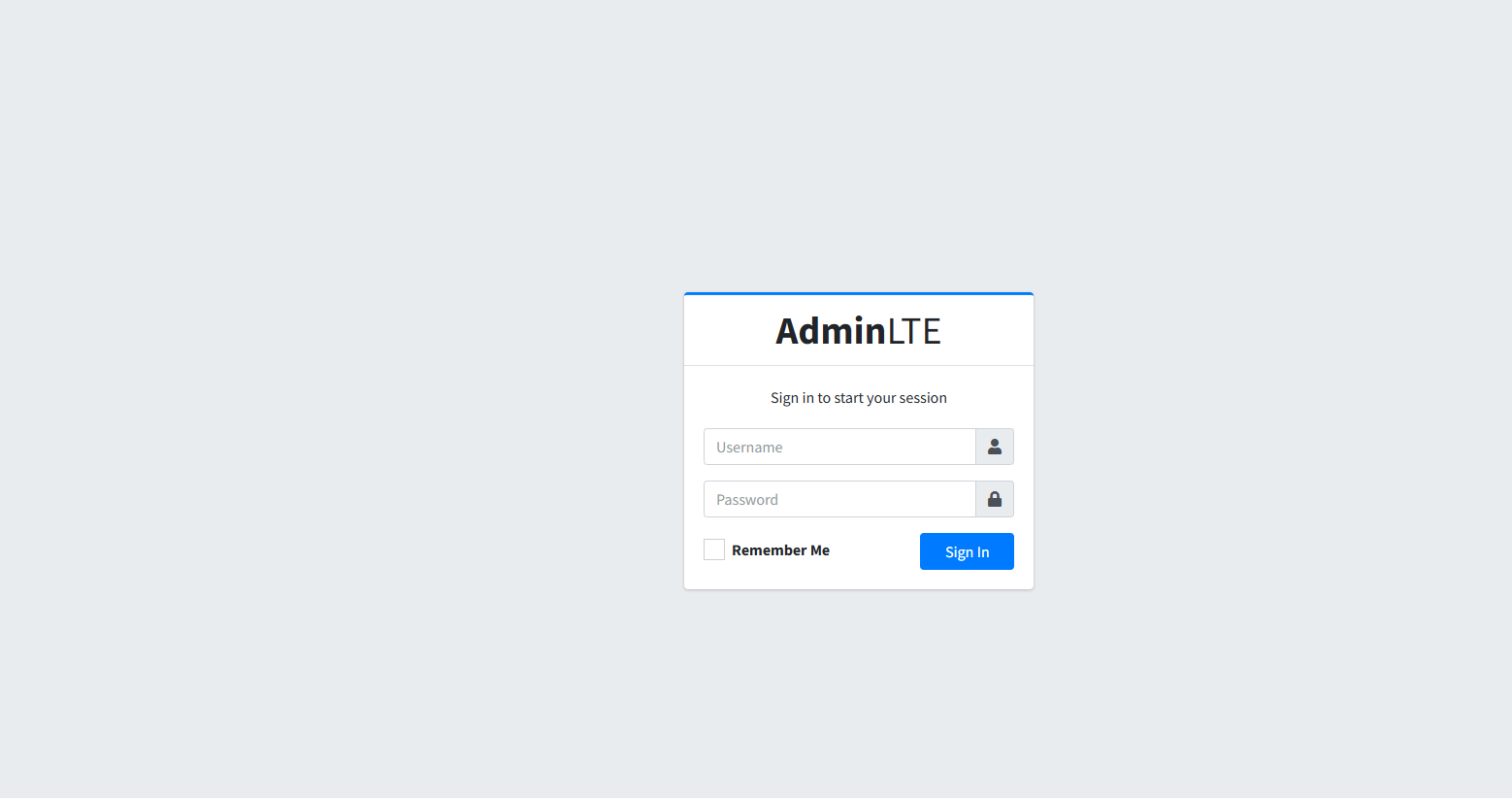
1. Selanjutnya kita coba jalankan pada browser dengan mengetikkan url localhost/dasarWeb/JS13.
2. Silahkan lakukan proses login dengan akun username : admin password : 12345
3. Amati apa yang terjadi, dan laporkan dalam laporan.

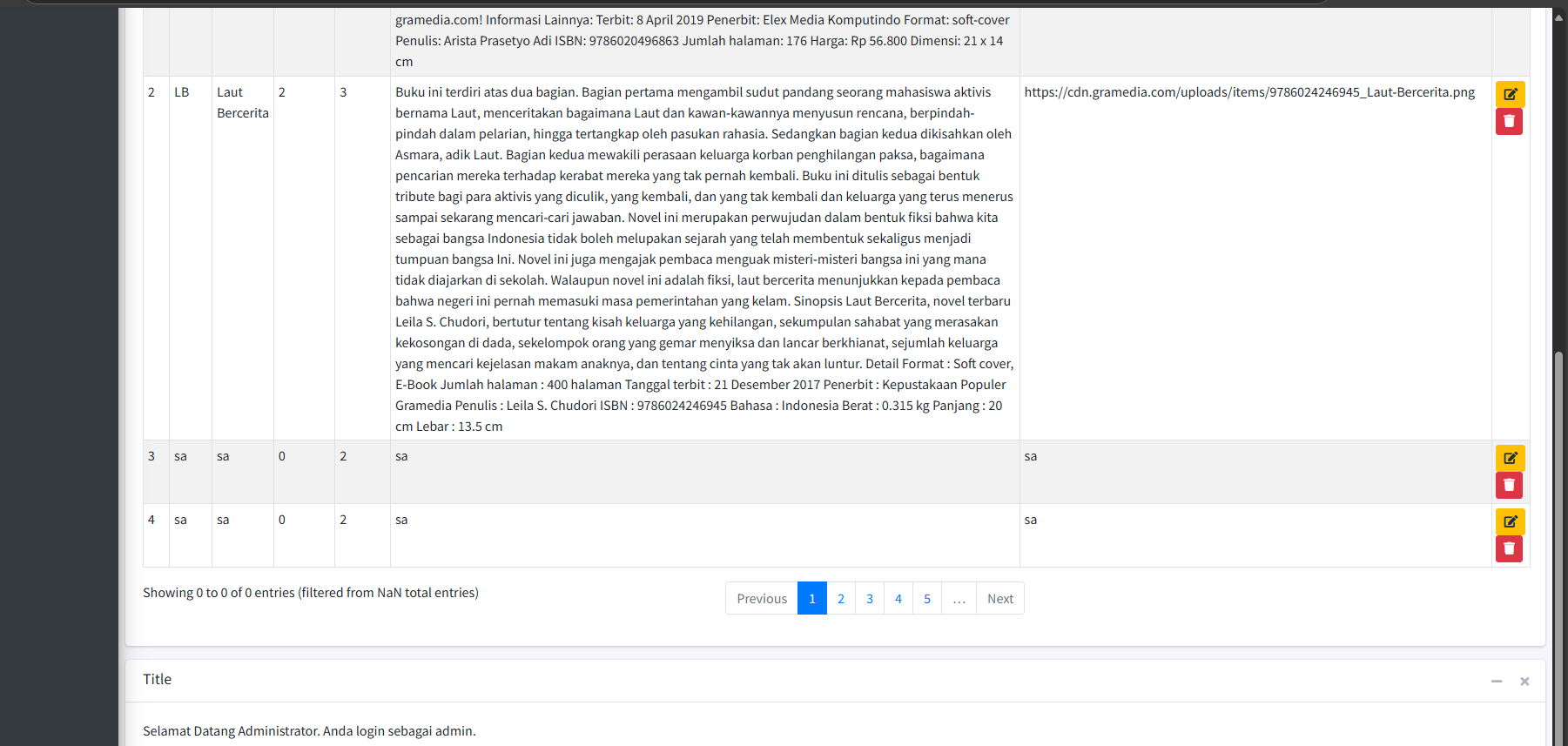
## Pertanyaan 2

Apa yang terjadi ketika menjalankan browser pada kode program yang sudah dibuat?

Screenshot hasil ujicoba pada browser dan jelaskan pendapatmu.

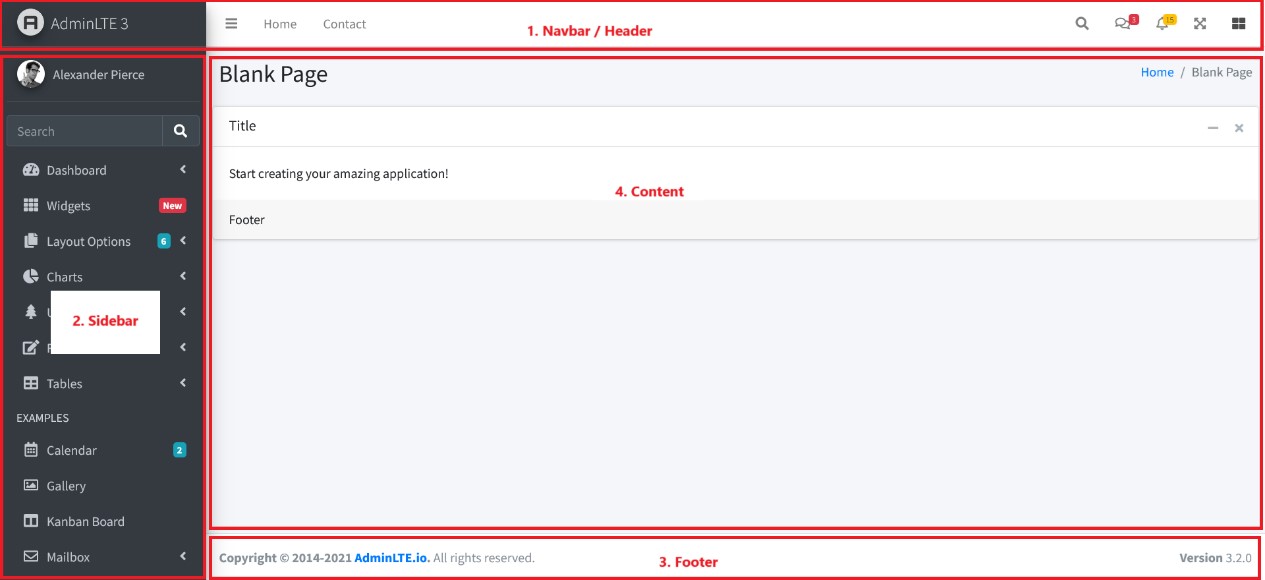






**Praktikum 2 - Layouting AdminLTE:**

1. Dari file index.php yang sudah kita buat, akan kita gunakan untuk proses layouting web
2. Kita lakukan layouting halaman index.php ini menjadi 5 element seperti pada gambar berikut



1. Kita edit file index.php dan kita modifikasi untuk memecah file tersebut menjadi 4 elemen, yaitu *header, sidebar, breadcrumb, content,* dan *footer*.
2. Kita buat elemen pertama, yaitu ***header***.
3. Kita ***cut*** kode diantara tanda <!-- Navbar --> dan <!-- /.navbar --> pada file index.php, kemudian kita ***paste***-kan pada file baru

C:\laragon\www\dasarWeb\JS13\layout\header.php

1. Selanjutnya kita load file layouts/header.php pada index.php seperti berikut



1. Selanjutnya, kita buat elemen kedua yaitu ***sidebar***.
2. Kita ***cut*** kode diantara tanda <!-- Sidebar --> dan <!-- /.sidebar --> pada file index.php, kemudian kita ***paste***-kan pada file baru

C:\laragon\www\dasarWeb\JS13\layout\sidebar.php

1. Selanjutnya kita load file layouts/sidebar.php pada index.php seperti berikut



1. Selanjutnya, kita buat elemen ketiga yaitu ***footer***.
2. Kita pindahkan kode pada tag <footer></footer> pada index.php ke file baru

C:\laragon\www\dasarWeb\JS13\layout\footer.php



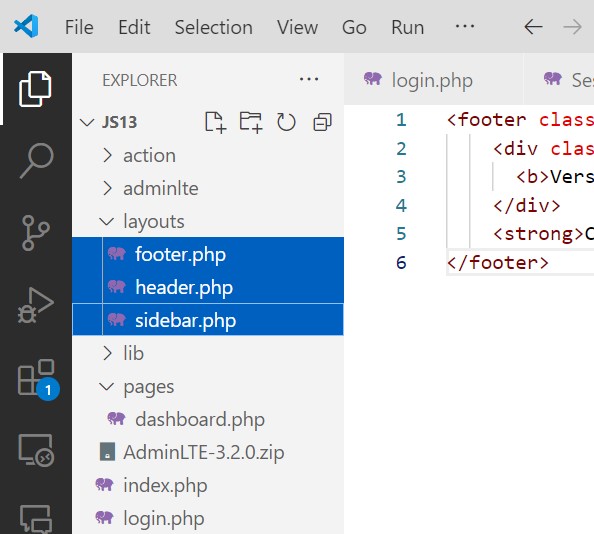
1. Selanjutnya kita load file layouts/footer.php pada index.php seperti berikut



1. Sehingga dalam folder layouts, terdapat 4 file yaitu header.php, sidebar.php, dan

footer.php

.



1. Sekarang kita buat layouts untuk content yang nantinya akan sering berganti-ganti sesuai dengan menu yang akses.
2. Kita modif file index.php, kita ***cut*** kode diatara tag

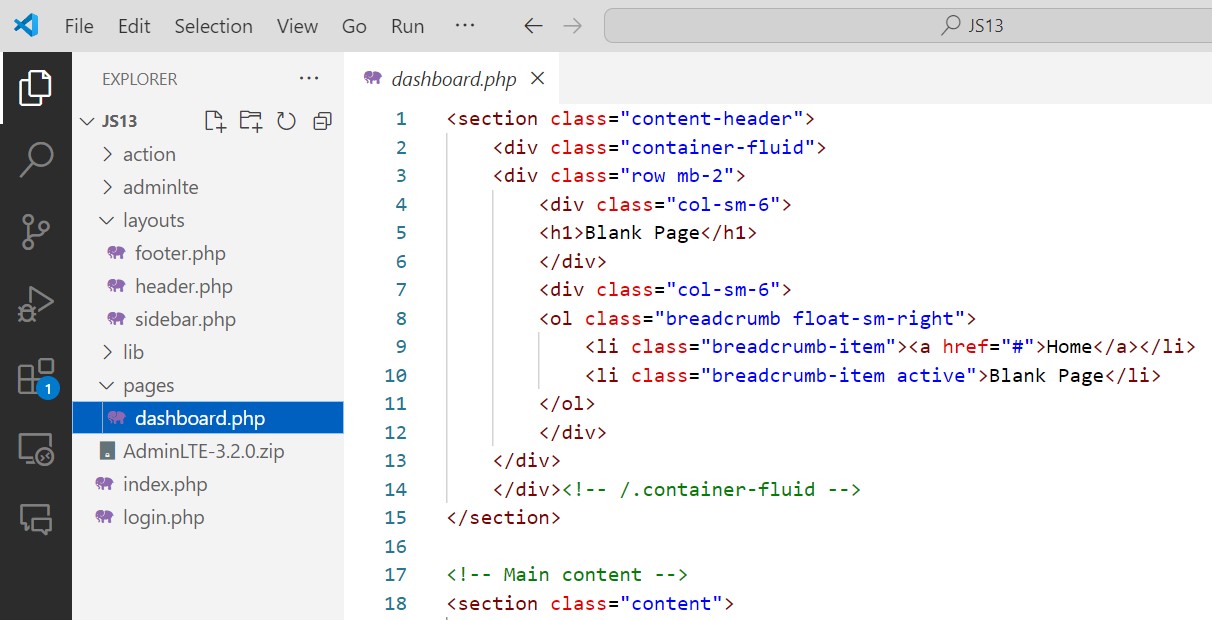
<!-- Content Header (Page header) --> dan <!-- /.content -->

Lalu kita paste-kan di file dan folder baru pada

C:\laragon\www\dasarWeb\JS13\pages\dashboard.php

16.

Kemudian diatara tag



<!-- Content Header (Page header) --> dan <!-- /.content --> Pada index.php, kita ubah menjadi seperti berikut



* 1. Ok, kita sudah berhasil melakukan layouting web template AdminLTE
  2. Sekarang silahkan dicoba dijalankan di browser dan amati apa yang terjadi

**Pertanyaan**

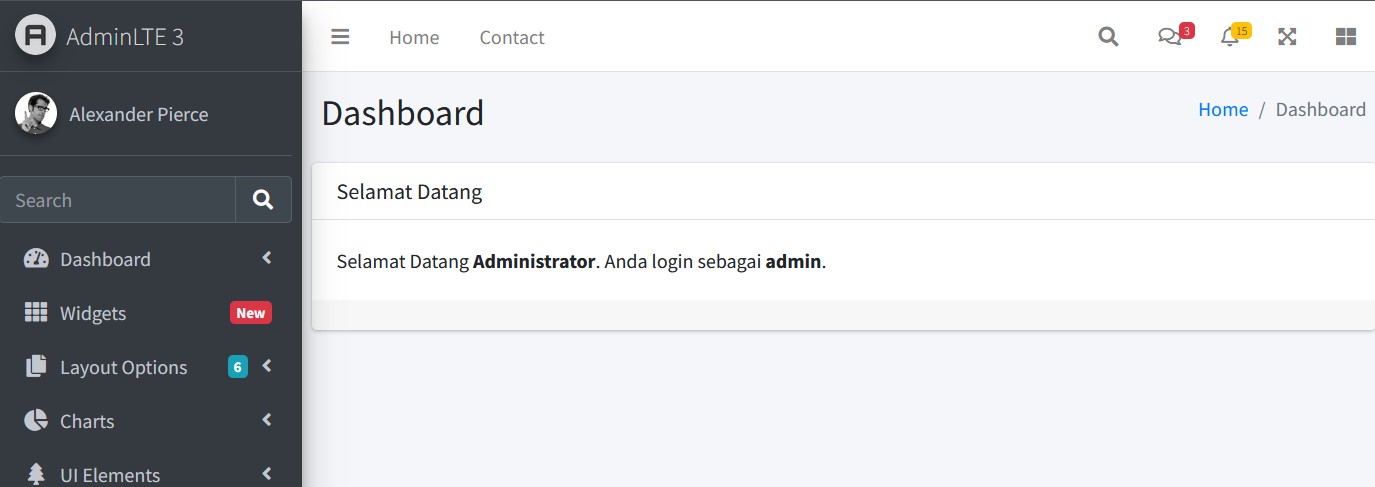
**3**

Silahkan ubah tampilan dashboard agar menampilkan nama pengguna yang telah

login

seperti gambar berikut

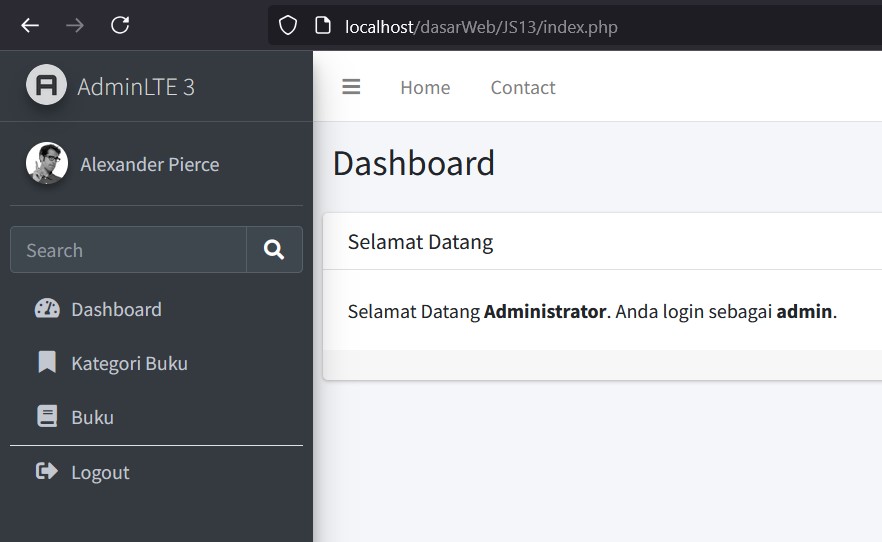
Screenshot hasil ujicoba pada browser dan jelaskan pendapatmu.



**Praktikum 3 - Penerapan Layouting:**

Sekarang kita akan mencoba melakukan penerapan terhadap layouting yang sudah kita lakukan.

1. Kita buat tampilan menu sidebar sesuai seperti pada gambar berikut



1. Kita modifikasi file layouts/sidebar.php

|  |
| --- |
| <div class="sidebar">  <!-- Sidebar user (optional) -->  <div class="user-panel mt-3 pb-3 mb-3 d-flex">  <div class="image">  <img src="adminlte/dist/img/user2-160x160.jpg" class="img-circle elevation-2" alt="User Image">  </div>  <div class="info">  <a href="#" class="d-block">Alexander Pierce</a>  </div>  </div>    <!-- SidebarSearch Form -->  <div class="form-inline">  <div class="input-group" data-widget="sidebar-search">  <input class="form-control form-control-sidebar" type="search"  placeholder="Search" aria-label="Search">  <div class="input-group-append"><button class="btn btn-sidebar"><i class="fas fa-search fa-fw"></i></button></div>  </div>  </div>    <!-- Sidebar Menu -->  <nav class="mt-2">  <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">  <!-- Add icons to the links using the .nav-icon class with font-awesome or any other icon font library -->  <li class="nav-item">  <a href="index.php" class="nav-link">  <i class="nav-icon fas fa-tachometer-alt"></i>  <p>Dashboard</p>  </a>  </li>  <li class="nav-item"> |

<a href="index.php?page=kategori" class="nav-link">

<i class="nav-icon fas fa-bookmark"></i>

<p>Kategori Buku</p>

</a>

</li>

<li class="nav-item border-bottom">

<a href="index.php?page=buku" class="nav-link">

<i class="nav-icon fas fa-book"></i>

<p>Buku</p>

</a>

</li>

<li class="nav-item">

<a href="action/auth.php?act=logout" class="nav-link">

<i class="nav-icon fas fa-sign-out-alt"></i>

<p>Logout</p>

</a>

</li>

</ul>

</nav>

<!-- /.sidebar-menu -->

</div>

1. Kita akan menerapkan jQuery Datatables seperti yang ada pada contoh di template

AdminLTE localhost/dasarWeb/JS13/adminlte/pages/tables/data.html, maka kita perlu memodifikasi file index.php .

1. Kita tambahkan kode untuk load css eksternal pada index.php. Kita tambahkan kode berikut, sebelum baris load css eksternal adminlte.min.css

<!-- DataTables -->

<link rel="stylesheet" href="adminlte/plugins/datatablesbs4/css/dataTables.bootstrap4.min.css">

<link rel="stylesheet" href="adminlte/plugins/datatablesresponsive/css/responsive.bootstrap4.min.css">

<link rel="stylesheet" href="adminlte/plugins/datatables-

buttons/css/buttons.bootstrap4.min.css">

Selanjutnya, kita pindahkan load library javascript jQuery yang ada di posisi bawah, ke posisi atas setelah load css eksternal adminlte.min.css (khusus untuk materi pembelajaran kali ini)



Maka tampilannya untuk bagian atas index.php akan seperti berikut



5. Kemudian, kita tambahkan load library javascript eksternal pada index.php (barisbaris terakhir) sebelum baris load javascript adminlte.min.js .

<!-- jQuery Validation -->

<script src="adminlte/plugins/jquery-validation/jquery.validate.min.js"></script>

<script src="adminlte/plugins/jquery-validation/additional-methods.min.js"></script>

<script src="adminlte/plugins/jquery-validation/localization/messages\_id.min.js"></script>

<!-- DataTables & Plugins -->

<script src="adminlte/plugins/datatables/jquery.dataTables.min.js"></script>

<script src="adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js"></script>

<script src="adminlte/plugins/datatables-

responsive/js/dataTables.responsive.min.js"></script>

<script src="adminlte/plugins/datatables-

responsive/js/responsive.bootstrap4.min.js"></script>

<script src="adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js"></script>

<script src="adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js"></script>

<script src="adminlte/plugins/jszip/jszip.min.js"></script>

<script src="adminlte/plugins/pdfmake/pdfmake.min.js"></script>

<script src="adminlte/plugins/pdfmake/vfs\_fonts.js"></script>

<script src="adminlte/plugins/datatables-buttons/js/buttons.html5.min.js"></script>

<script src="adminlte/plugins/datatables-buttons/js/buttons.print.min.js"></script> <script src="adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js"></script>

Maka akan tampil kode seperti gambar berikut

6.

Kemudian, kita buat file baru di



C:\laragon\www\dasarWeb\JS13\pages\kategori.php dan kita ketikkan kode berikut

<section class="content-header">

<div class="container-fluid">

<div class="row mb-2">

<div class="col-sm-6">

<h1>Kategori Buku</h1>

</div>

<div class="col-sm-6">

<ol class="breadcrumb float-sm-right">

<li class="breadcrumb-item"><a href="#">Home</a></li>

<li class="breadcrumb-item active">Kategori</li>

</ol>

</div>

</div>

</div>

</section>

<!-- Main content -->

<section class="content">

<div class="card">

<div class="card-header">

<h3 class="card-title">Daftar Kategori Buku</h3>

<div class="card-tools">

<button type="button" class="btn btn-md btn-primary" onclick="tambahData()">

Tambah

</button>

</div>

</div>

<div class="card-body">

<table class="table table-sm table-bordered table-striped" id="table-data">

<thead>

<tr>

<th>No</th>

|  |
| --- |
| <th>Kode Kategori</th>  <th>Nama Kategori</th>  <th>Aksi</th>  </tr>  </thead> <tbody>  </tbody>  </table>  </div>  </div>  </section>  <div class="modal fade" id="form-data" style="display: none;" aria-hidden="true">  <form action="action/kategoriAction.php?act=save" method="post" id="form-tambah">  <!-- Ukuran Modal  modal-sm : Modal ukuran kecil modal-md : Modal ukuran sedang modal-lg : Modal ukuran besar modal-xl : Modal ukuran sangat besar  penerapan setelah class modal-dialog seperti di bawah -->  <div class="modal-dialog modal-md">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title">Tambah Kategori</h4>  </div>  <div class="modal-body">  <div class="form-group">  <label>Kode Kategori</label>  <input type="text" class="form-control" name="kategori\_kode" id="kategori\_kode"> </div>  <div class="form-group">  <label>Nama Kategori</label>  <input type="text" class="form-control" name="kategori\_nama" id="kategori\_nama"> </div>  </div>  <div class="modal-footer justify-content-between"> <button type="button" class="btn btn-default" datadismiss="modal">Batal</button>  <button type="submit" class="btn btn-primary">Simpan</button>  </div>  </div>  </div>  </form>  </div>    <script>  function tambahData(){  $('#form-data').modal('show');  $('#form-tambah').attr('action', 'action/kategoriAction.php?act=save');  $('#kategori\_kode').val('');  $('#kategori\_nama').val('');  }  function editData(id){  $.ajax({  url: 'action/kategoriAction.php?act=get&id='+id, method: 'post',  success: function(response){ var data = JSON.parse(response);  $('#form-data').modal('show');  $('#form-tambah').attr('action',  'action/kategoriAction.php?act=update&id='+id);  $('#kategori\_kode').val(data.kategori\_kode);  $('#kategori\_nama').val(data.kategori\_nama); |

|  |
| --- |
| }  });  }  function deleteData(id){  if(confirm('Apakah anda yakin?')){  $.ajax({  url: 'action/kategoriAction.php?act=delete&id='+id, method: 'post',  success: function(response){ var result = JSON.parse(response); if(result.status){  tabelData.ajax.reload();  }else{  alert(result.message);  }  }  });  }  }  var tabelData;  $(document).ready(function(){  tabelData = $('#table-data').DataTable({ ajax: 'action/kategoriAction.php?act=load',  });    $('#form-tambah').validate({ rules: {  kategori\_kode: { required: true, minlength: 3  },  kategori\_nama: { required: true, minlength: 5  } },  errorElement: 'span',  errorPlacement: function(error, element){ error.addClass('invalid-feedback');  element.closest('.form-group').append(error);  },  highlight: function(element, errorClass, validClass){  $(element).addClass('is-invalid');  },  unhighlight: function(element, errorClass, validClass){  $(element).removeClass('is-invalid');  },  submitHandler: function(form){  $.ajax({  url: $(form).attr('action'),  method: 'post',  data: $(form).serialize(), success: function(response){ var result = JSON.parse(response); if(result.status){  $('#form-data').modal('hide');  tabelData.ajax.reload(); // reload data tabel  }else{  alert(result.message);  }  }  });  }  }); |

});

</script>

1. Selanjutnya kita buat file baru di C:\laragon\www\dasarWeb\JS13\lib\Secure.php Kemudian kita ketikkan kode berikut

|  |
| --- |
| <?php  function antiSqlInjection($data){ $data = stripslashes($data); $data = strip\_tags($data);  $data = htmlentities($data);  $data = htmlspecialchars($data); $data = addslashes($data); return $data;  } |

1. Selanjutnya kitabuat folder baru model, dan kita buat file baru di dalamnya dengan nama

C:\laragon\www\dasarWeb\JS13\model\Model.php. File Model.php ini adalah abstract class untuk class model yang lain.

<?php

abstract class Model{

public abstract function insertData($id); public abstract function getData(); public abstract function getDataById($id); public abstract function updateData($id, $data); public abstract function deleteData($id); }

1. Kemudian kita juga buat file KategoriModel.php di dalam folder model

|  |
| --- |
| <?php  include('Model.php');    class KategoriModel extends Model{  private $db;  private $table = 'm\_kategori';    public function \_\_construct(){  include\_once('../lib/Connection.php');    $this->db = $db;  $this->db->set\_charset('utf8');  }  public function insertData($data){  // prepare statement untuk query insert  $query = $this->db->prepare("insert into {$this->table} (kategori\_kode, kategori\_nama) values(?,?)");    // binding parameter ke query, "s" berarti string, "ss" berarti dua string  $query->bind\_param('ss', $data['kategori\_kode'], $data['kategori\_nama']);  // eksekusi query untuk menyimpan ke database  $query->execute();  } |
| public function getData(){  // query untuk mengambil data dari tabel bank\_soal return $this->db->query("select \* from {$this->table} ");  }  public function getDataById($id){    // query untuk mengambil data berdasarkan id  $query = $this->db->prepare("select \* from {$this->table} where kategori\_id = ?");  // binding parameter ke query "i" berarti integer. Biar tidak kena SQL Injection  $query->bind\_param('i', $id);    // eksekusi query  $query->execute();    // ambil hasil query  return $query->get\_result()->fetch\_assoc();  }  public function updateData($id, $data){  // query untuk update data  $query = $this->db->prepare("update {$this->table} set kategori\_kode = ?, kategori\_nama = ? where kategori\_id = ?");    // binding parameter ke query  $query->bind\_param('ssi', $data['kategori\_kode'], $data['kategori\_nama'], $id);    // eksekusi query  $query->execute();  }  public function deleteData($id){  // query untuk delete data  $query = $this->db->prepare("delete from {$this->table} where kategori\_id = ?");  // binding parameter ke query  $query->bind\_param('i', $id);    // eksekusi query  $query->execute();  }  } |

1. Bagian akhir, kita bisa membuat file controller yang memproses CRUD data. Kita buat file kategoriAction.php pada folder action.

C:\laragon\www\dasarWeb\JS13\action\kategoriAction.php

|  |
| --- |
| <?php  include('../lib/Session.php');    $session = new Session();    if($session->get('is\_login') !== true){ header('Location: login.php');  }  include\_once('../model/KategoriModel.php'); include\_once('../lib/Secure.php');    $act = isset($\_GET['act'])? strtolower($\_GET['act']) : '';    if($act == 'load'){ |

|  |
| --- |
| $kategori = new KategoriModel();  $data = $kategori->getData();  $result = []; $i = 1;  while($row = $data->fetch\_assoc()){  $result['data'][] = [  $i,  $row['kategori\_kode'],  $row['kategori\_nama'],  '<button class="btn btn-sm btn-warning"  onclick="editData('.$row['kategori\_id'].')"><i class="fa fa-edit"></i></button>  <button class="btn btn-sm btn-danger"  onclick="deleteData('.$row['kategori\_id'].')"><i class="fa fa-trash"></i></button>'  ];  $i++;  }  echo json\_encode($result);  }  if($act == 'get'){  $id = (isset($\_GET['id']) && ctype\_digit($\_GET['id']))? $\_GET['id'] : 0;  $kategori = new KategoriModel(); $data = $kategori->getDataById($id); echo json\_encode($data);  }  if($act == 'save'){  $data = [  'kategori\_kode' => antiSqlInjection($\_POST['kategori\_kode']),  'kategori\_nama' => antiSqlInjection($\_POST['kategori\_nama'])  ];    $kategori = new KategoriModel();  $kategori->insertData($data);    echo json\_encode([ 'status' => true,  'message' => 'Data berhasil disimpan.'  ]);  }  if($act == 'update'){  $id = (isset($\_GET['id']) && ctype\_digit($\_GET['id']))? $\_GET['id'] : 0; $data = [  'kategori\_kode' => antiSqlInjection($\_POST['kategori\_kode']),  'kategori\_nama' => antiSqlInjection($\_POST['kategori\_nama'])  ];    $kategori = new KategoriModel();  $kategori->updateData($id, $data);    echo json\_encode([ 'status' => true,  'message' => 'Data berhasil diupdate.'  ]);  }  if($act == 'delete'){  $id = (isset($\_GET['id']) && ctype\_digit($\_GET['id']))? $\_GET['id'] : 0;  $kategori = new KategoriModel();  $kategori->deleteData($id);    echo json\_encode([ 'status' => true, |

'message' => 'Data berhasil dihapus.'

]);

}

1. Sekarang kita coba jalankan di browser, dan mengakses menu **Kategori Buku.**  Kita coba lakukan proses CRUD

<http://localhost/dasarWeb/JS13>

## Pertanyaan 4

Silahkan lakukan analisis terhadap file-file yang telah dibuat pada Praktikum 3 ini.

Jabarkan masing-masing file yang dibuat, dan jelaksan maksud dari file-file tersebut.

**1. index.php**

* **Deskripsi**: Ini adalah file utama yang berfungsi sebagai titik masuk aplikasi. File ini biasanya menangani routing dan menampilkan konten berdasarkan parameter yang diterima (misalnya, **page**).
* **Maksud**: Mengarahkan pengguna ke halaman yang sesuai berdasarkan permintaan mereka. Jika pengguna mengakses **index.php?page=kategori**, maka file ini akan memuat konten dari **kategori.php**.

**2. lib/Session.php**

* **Deskripsi**: File ini berisi kelas **Session** yang mengelola sesi pengguna. Kelas ini biasanya memiliki metode untuk memulai sesi, menyimpan data sesi, dan mengambil data sesi.
* **Maksud**: Mengelola status login pengguna dan menyimpan informasi penting selama sesi, seperti apakah pengguna sudah login atau tidak.

**3. layouts/header.php**

* **Deskripsi**: File ini berisi kode HTML untuk bagian header dari aplikasi, termasuk navbar dan elemen navigasi lainnya.
* **Maksud**: Menyediakan antarmuka pengguna yang konsisten di seluruh halaman dengan menampilkan header yang sama.

**4. layouts/sidebar.php**

* **Deskripsi**: File ini berisi kode HTML untuk sidebar aplikasi, yang biasanya berisi menu navigasi.
* **Maksud**: Memudahkan pengguna untuk menavigasi ke berbagai bagian aplikasi, seperti dashboard, kategori, buku, dan pengguna.

**5. layouts/footer.php**

* **Deskripsi**: File ini berisi kode HTML untuk bagian footer dari aplikasi.
* **Maksud**: Menyediakan informasi tambahan, seperti hak cipta dan versi aplikasi, di bagian bawah setiap halaman.

**6. pages/dashboard.php**

* **Deskripsi**: File ini berisi konten untuk halaman dashboard aplikasi.
* **Maksud**: Menyediakan ringkasan informasi penting dan statistik yang relevan bagi pengguna setelah mereka login.

**7. pages/kategori.php**

* **Deskripsi**: File ini berisi kode untuk menampilkan dan mengelola kategori buku.
* **Maksud**: Menyediakan antarmuka untuk menambah, mengedit, dan menghapus kategori buku. Ini juga berfungsi untuk menampilkan daftar kategori yang ada.

**8. pages/buku.php**

* **Deskripsi**: File ini berisi kode untuk menampilkan dan mengelola data buku.
* **Maksud**: Menyediakan antarmuka untuk menambah, mengedit, dan menghapus buku. Ini juga berfungsi untuk menampilkan daftar buku yang ada.

**9. pages/user.php**

* **Deskripsi**: File ini berisi kode untuk mengelola pengguna dalam aplikasi.
* **Maksud**: Menyediakan antarmuka untuk menambah, mengedit, dan menghapus pengguna, serta menampilkan daftar pengguna yang ada.

**10. pages/404.php**

* **Deskripsi**: File ini berisi kode untuk menampilkan halaman kesalahan 404 ketika halaman yang diminta tidak ditemukan.
* **Maksud**: Memberikan umpan balik kepada pengguna bahwa halaman yang mereka cari tidak ada, dan memberikan opsi untuk kembali ke halaman sebelumnya atau ke halaman utama.

**11. action/kategoriAction.php**

* **Deskripsi**: File ini berisi logika untuk menangani aksi terkait kategori, seperti menambah, mengedit, menghapus, dan memuat data kategori.
* **Maksud**: Mengelola interaksi dengan database untuk kategori buku, termasuk validasi dan pengolahan data yang diterima dari formulir.

**12. Model.php**

* **Deskripsi**: File ini berisi kelas abstrak **Model** yang mendefinisikan metode dasar untuk operasi CRUD (Create, Read, Update, Delete).
* **Maksud**: Menyediakan struktur dasar untuk model-model lain yang akan mengimplementasikan metode ini, seperti **KategoriModel**.

**13. model/KategoriModel.php**

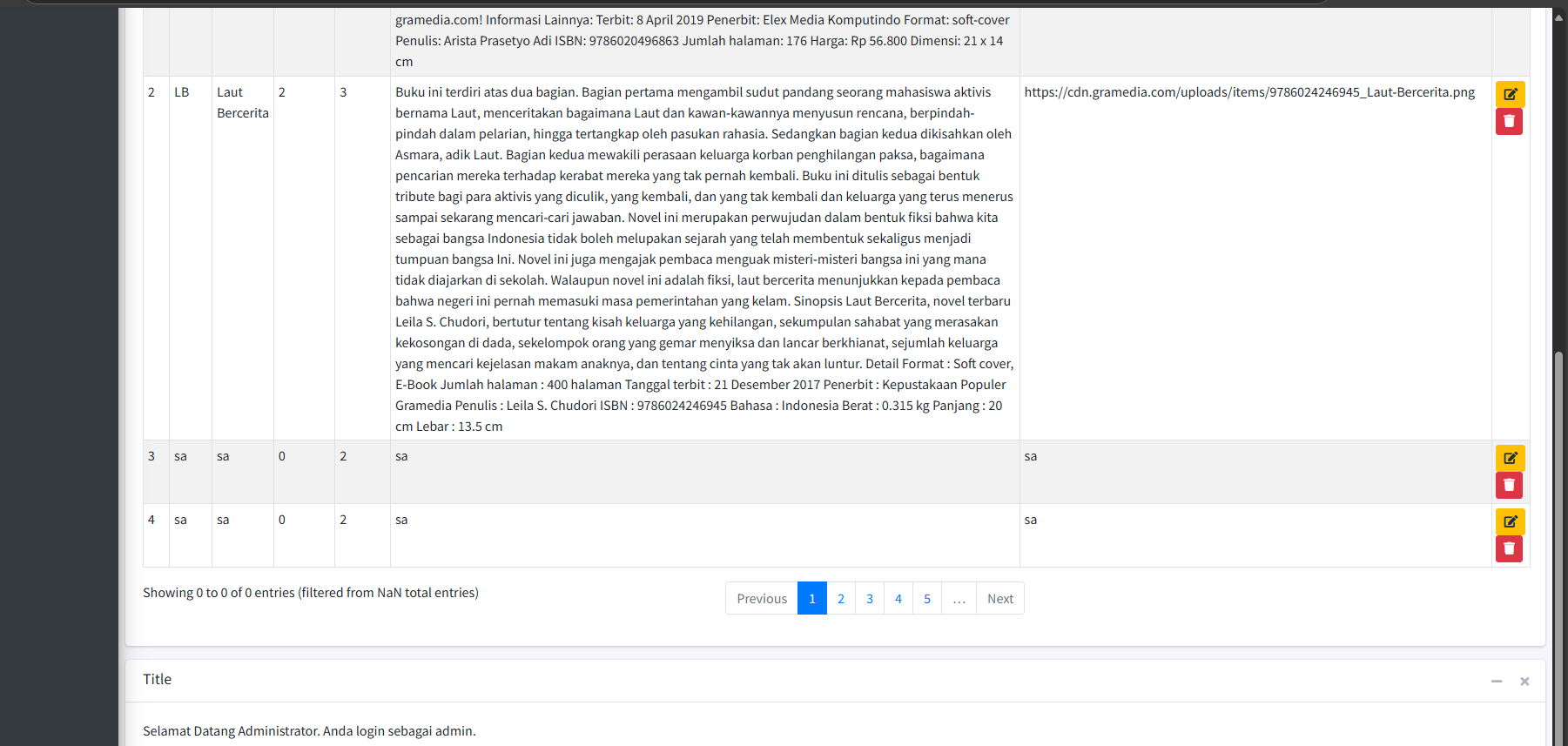
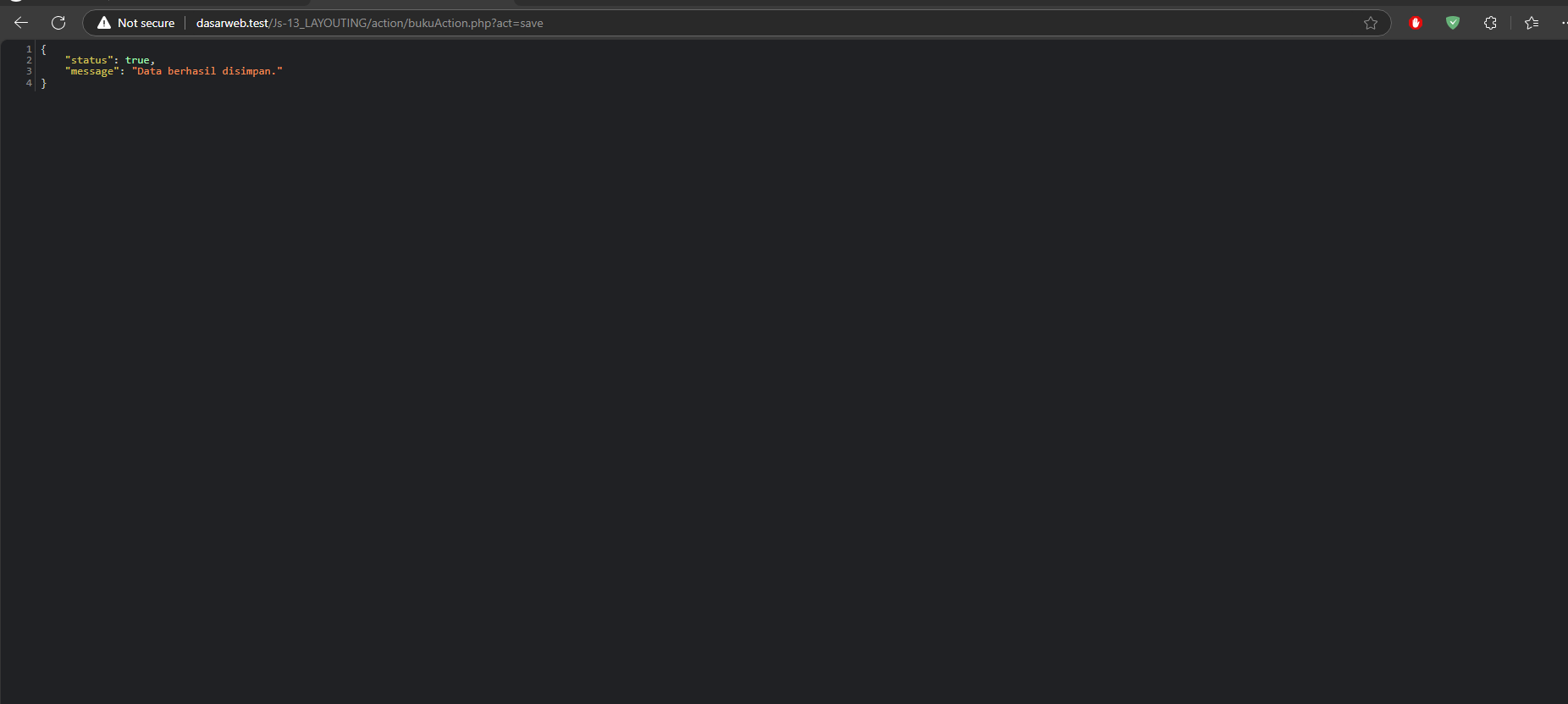
* **Deskripsi**: File ini berisi kelas **KategoriModel** yang mengimplementasikan metode dari kelas **Model** untuk mengelola data kategori.
* **Maksud**: Menyediakan logika spesifik untuk berinteraksi dengan tabel kategori di database, termasuk metode untuk menambah, mengedit, dan menghapus kategori.

**14. lib/Connection.php**

* **Deskripsi**: File ini berisi kode untuk mengatur koneksi ke database.
* **Maksud**: Menyediakan koneksi yang diperlukan untuk berinteraksi dengan database, sehingga model dapat melakukan operasi CRUD.

## Pertanyaan 5

Silahkan implementasikan kode yang sudah dibuat untuk membuat CRUD menu Buku.



*\*\*\* Sekian, dan selamat belajar \*\*\**