



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1. Kedudukan dan Koordinasi**

Kerja magang dilaksanakan pada divisi *web developer intern*, yang terdiri atas sebuah tim dengan lima orang anggota. Dalam mengerjakan pengembangan situs, tim dibantu oleh seorang pembimbing lapangan, yaitu Bapak Nurdiansyah, S.Kep., Ns., selaku CEO dan *founder* Aveecana. Bapak Nurdiansyah berperan membantu tim perihal perencanaan proyek, penentuan *user requirements*, penentuan ekspektasi hasil akhir situs, serta pemberian masukan serta referensi kepada tim selama pengembangan situs dilakukan.

#### **3.2. Tugas yang Dilakukan**

Selama dua bulan pelaksanaan kerja magang, secara garis besar, tugas utama yang diemban adalah mengembangkan *front-end* dari sebuah situs pembelajaran *online* yang ditujukan bagi praktisi-praktisi kesehatan. Pada dua bulan tersebut, pengembangan fokus pada halaman bagi admin. Untuk menyelesaikan tugas tersebut, ada beberapa pencapaian atau *milestones* yang harus diselesaikan. Rangkaian *milestones* tersebut dirancang dalam *timeline* proyek sebagai berikut:

**Tabel 3.1 Timeline Pelaksanaan Kerja Magang**

Pekerjaan	Februari				Maret			
	Minggu ke-							
	1	2	3	4	5	6	7	8
Membuat perencanaan dan mempelajari referensi-referensi	■							
Menganalisis <i>requirements sistem</i>		■						
Mendesain interface dashboard user			■					
Mendesain interface dashboard admin				■				
Mengembangkan front-end dashboard admin (index, session, koneksi database)					■			
Mengembangkan front-end dashboard admin (laman profile, daftar admin)						■		
Mengembangkan front-end dashboard admin (laman chapter)							■	
Mengembangkan front-end dashboard admin (laman content)								■

3.2.1. Minggu Pertama: Membuat perencanaan dan mempelajari referensi-referensi

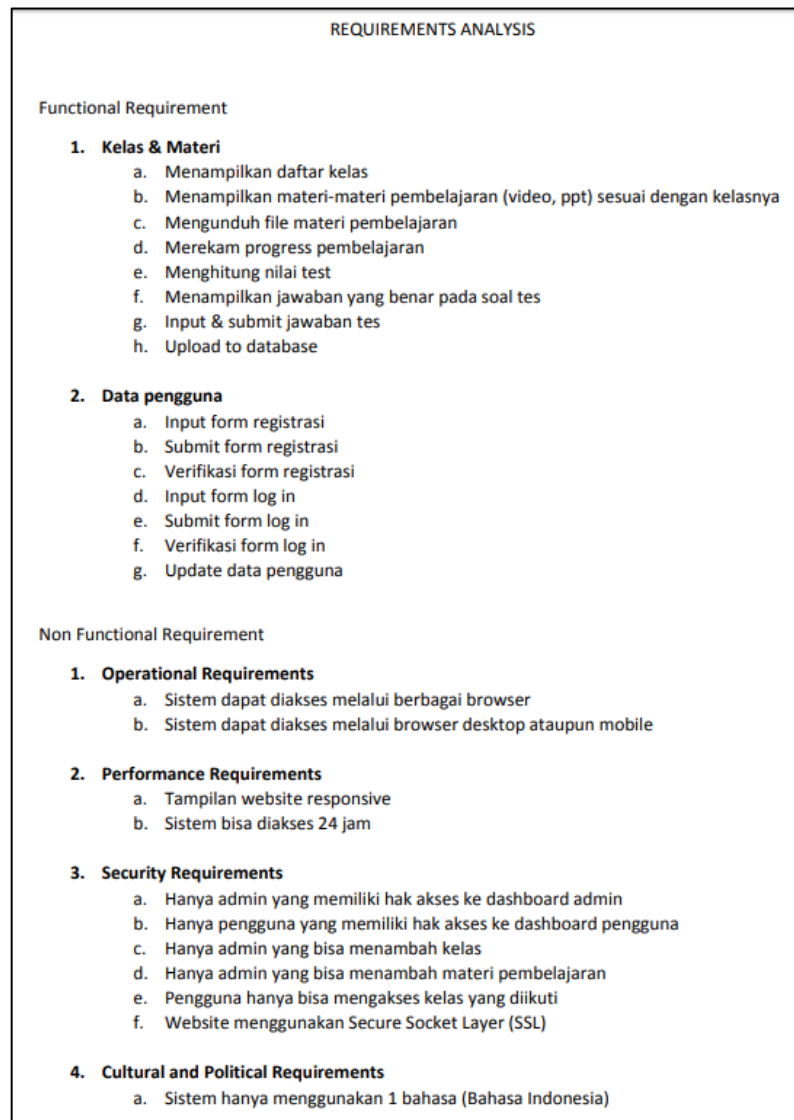
Di minggu pertama kerja magang, yang dilakukan adalah merencanakan proyek. Perencanaan mencakup pembagian *job description* tiap anggota tim, penentuan metode pengembangan sistem, pembuatan *timeline* pengerjaan, serta penentuan perangkat lunak yang akan digunakan saat mendesain dan mengembangkan sistem. Diskusi perencanaan ini menghasilkan keputusan untuk menggunakan *System Development Life Cycle* (SDLC) sebagai metode pengembangan sistem. SDLC terdiri dari enam fase, dimulai dari *preliminary, analysis, design, coding, testing, maintenance* (Navita, 2017). Selain itu, diputuskan juga untuk menggunakan *software Adobe XD* untuk merancang *user interface*.

Selain perencanaan, pembimbing juga memberikan beberapa referensi sistem serupa yang dikumpulkan dari hasil-hasil riset terdahulu perusahaan

seputar situs *e-course* ini. Referensi-referensi membantu tim mendapatkan gambaran sistem, tampilan sistem, serta daftar fitur yang diperlukan.

### 3.2.2. Minggu Kedua: Menganalisis *requirements* sistem

Pada minggu kedua, tugas yang dikerjakan adalah menganalisis dan menentukan *requirements* sistem. Tugas ini dilakukan dengan cara berdiskusi dengan pembimbing lapangan. Pembimbing lapangan mengutarakan fitur-fitur yang diperlukan, alur penggunaan sistem oleh admin maupun *user*, serta memberikan berbagai macam referensi. Dari hasil analisis dan diskusi tersebut, terbentuklah dokumen akhir *requirements analysis* seperti yang tertera pada gambar 3.1.

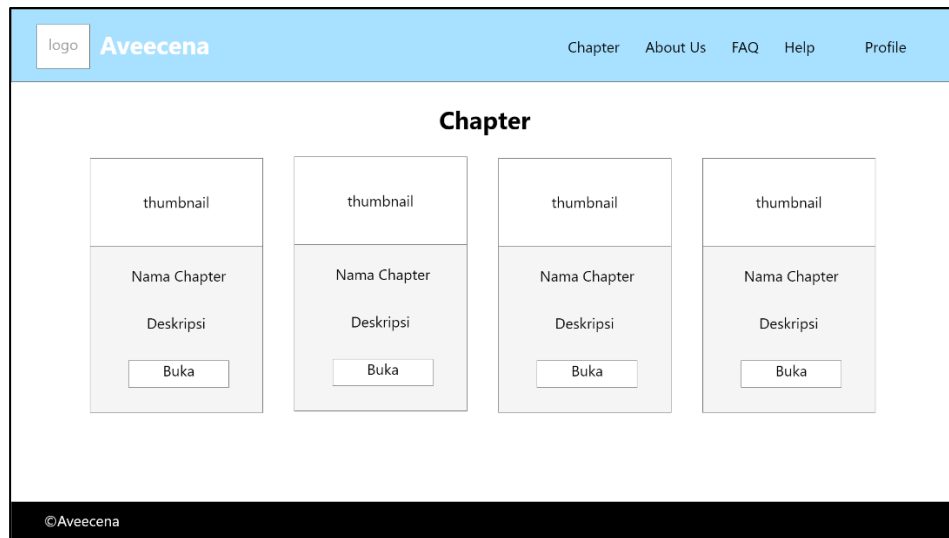


**Gambar 3.1 Requirements Analysis**

### 3.2.3. Minggu Ketiga: Mendesain *interface dashboard user*

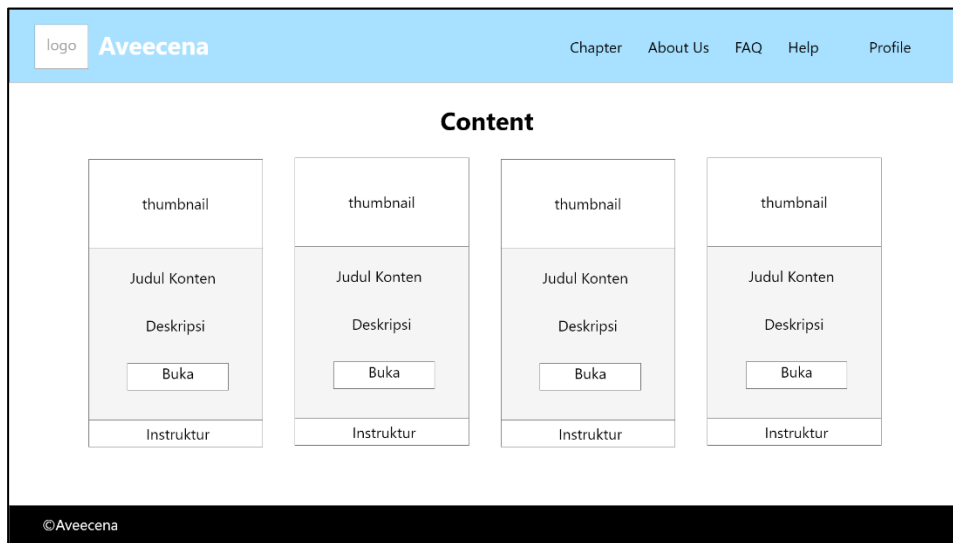
Setelah mengetahui *user requirements*, pada minggu berikutnya, tugas yang dilakukan adalah membuat *interface website* untuk *dashboard user*. *Interface* yang dibuat hanya menitikberatkan pada tata letak atau *layout*-nya sebagai pedoman saat pengembangan sistem. Mengenai pewarnaan, pemilihan *font*, dan rincian desain lainnya dilakukan secara langsung ketika

pengembangan sistem. *Layout interface website* dirancang menggunakan perangkat lunak *Adobe XD*. Berikut adalah hasil rancangan untuk tampilan *user*:



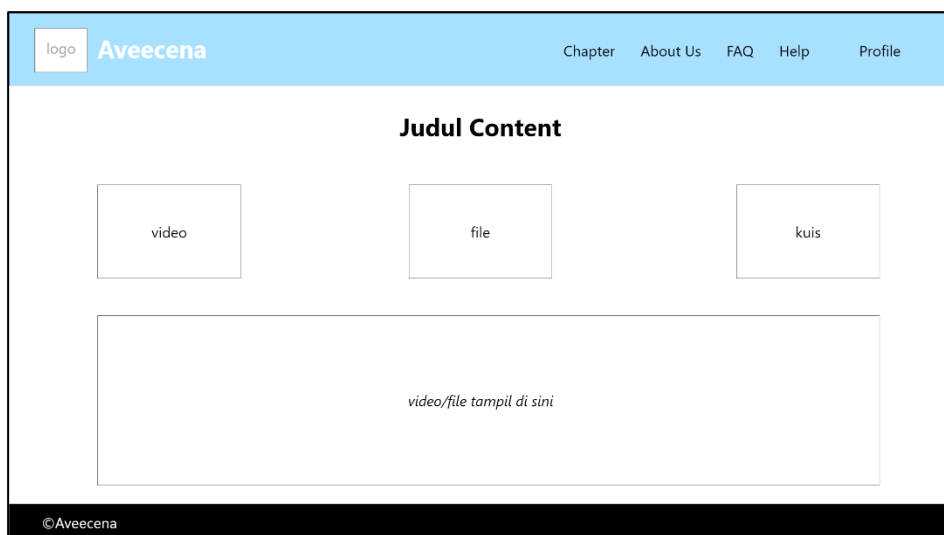
**Gambar 3. 2 UI Halaman Daftar *Chapter***

Setelah *user* melakukan *log in*, *user* akan diarahkan ke halaman *chapter*. Pada halaman ini, terdapat daftar *chapter* yang tersedia. Yang dimaksud dengan *chapter* adalah modul, di mana berisikan beberapa konten atau materi di dalamnya. Informasi yang ditampilkan terdiri *thumbnail chapter*, nama *chapter*, serta deskripsi singkat dari *chapter* tersebut. Jika ingin melihat konten-konten dari kelas tersebut, maka *user* perlu melakukan klik pada *button Buka*. Ada *chapter* yang gratis, namun ada pula yang berbayar. *Chapter* yang berbayar maka harga akan ditampilkan pada *button Buka* kemudian diarahkan ke halaman pembayaran.



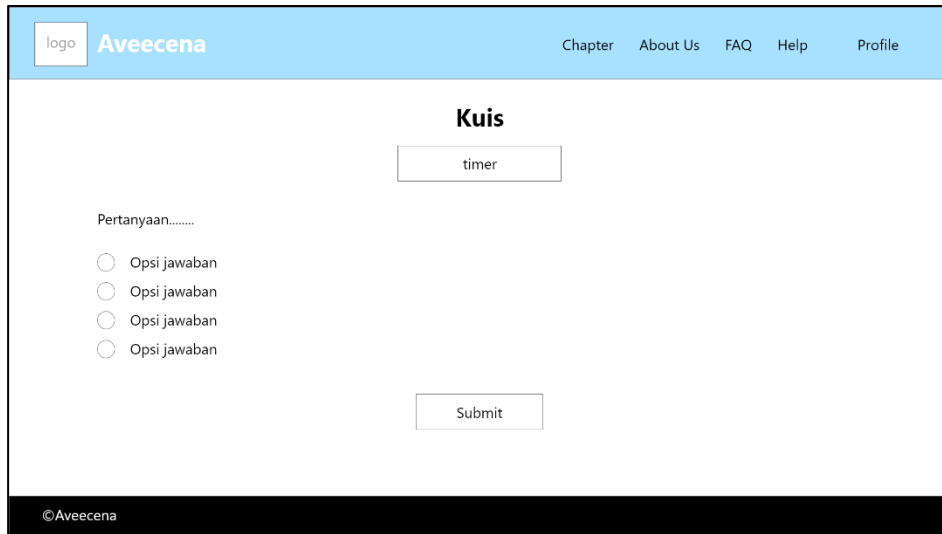
**Gambar 3. 3 UI Halaman Daftar Konten**

Setelah *user* melakukan klik pada *button Buka*, akan muncul daftar konten yang tersedia. Informasi yang ditampilkan yakni *thumbnail* konten, judul konten, deskripsi singkat konten, serta nama instruktur atau pengajar.



**Gambar 3. 4 UI Halaman Isi Konten**

Setelah *user* melakukan klik pada *button Buka* di daftar konten. akan muncul isi-isi atau materi dari konten tersebut. Materi dari setiap konten akan disajikan dalam bentuk video, *file*, dan kuis.



**Gambar 3. 5 UI Halaman Kuis**

Kuis akan disajikan berupa soal-soal pilihan ganda. Setiap kuis akan dibatasi dengan waktu yang juga dapat dilihat di layar. Di akhir kuis, setelah *user* melakukan *submit* kuis, *user* dapat langsung melihat nilai beserta kunci jawabannya.



logo Aveecena Chapter About Us FAQ Help Profile

### Pembayaran

No. Invoice: #####  
Nominal Tagihan: #####  
Dibayarkan ke:  
Nama & No Rekening

Upload Bukti Transfer

©Aveecena

**Gambar 3. 6 UI Halaman Pembayaran *Chapter***

Apabila *chapter* yang ingin diakses *user* sifatnya berbayar, maka *user* akan diarahkan ke halaman pembayaran. *User* akan diberitahu berapa nominal yang harus dibayarkan dan ke mana pembayaran harus dilakukan. Untuk sementara, pembayaran masih dilakukan secara manual melalui transfer *bank*. Setelah melakukan pembayaran, *user* harus mengisi formulir konfirmasi pembayaran di mana *user* harus mengunggah foto bukti pembayaran, nama bank yang digunakan untuk melakukan transfer, nama pemilik rekening, serta tanggal dilakukan transfer. Jika formulir semua sudah terisi, *user* klik *button Submit*, lalu sistem akan memberi pemberitahuan melalui *e-mail* apabila pembayaran sudah terkonfirmasi. Jika pembayaran sudah terkonfirmasi, akses untuk materi berbayar tersebut menjadi terbuka untuk *user* tersebut.

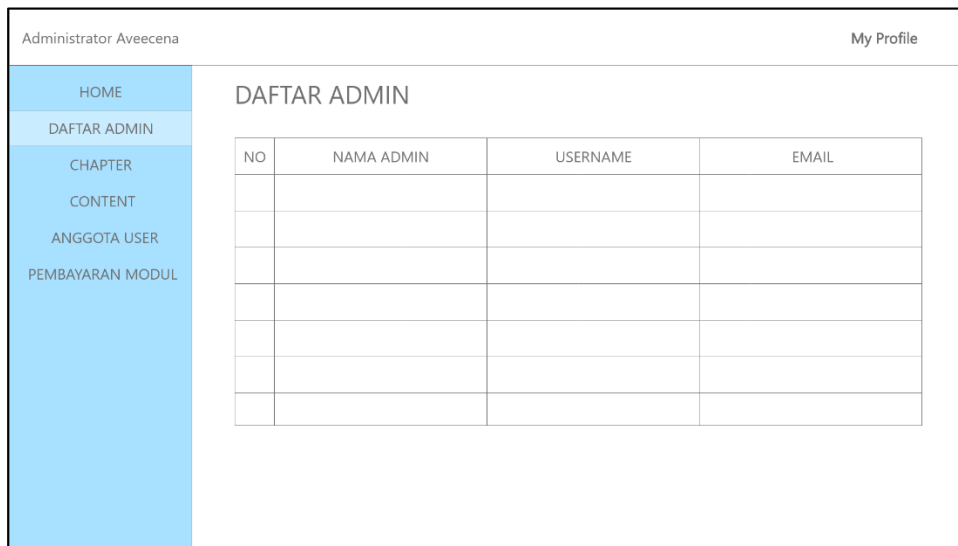
### 3.2.4. Minggu Keempat: Mendesain *interface website (dashboard admin)*

Setelah selesai merancang *interface dashboard user*, pada minggu berikutnya, tugas yang dilakukan merancang *interface dashboard admin*. *Layout interface website* juga dirancang menggunakan perangkat lunak *Adobe XD*. Berikut adalah hasil rancangan untuk tampilan admin:



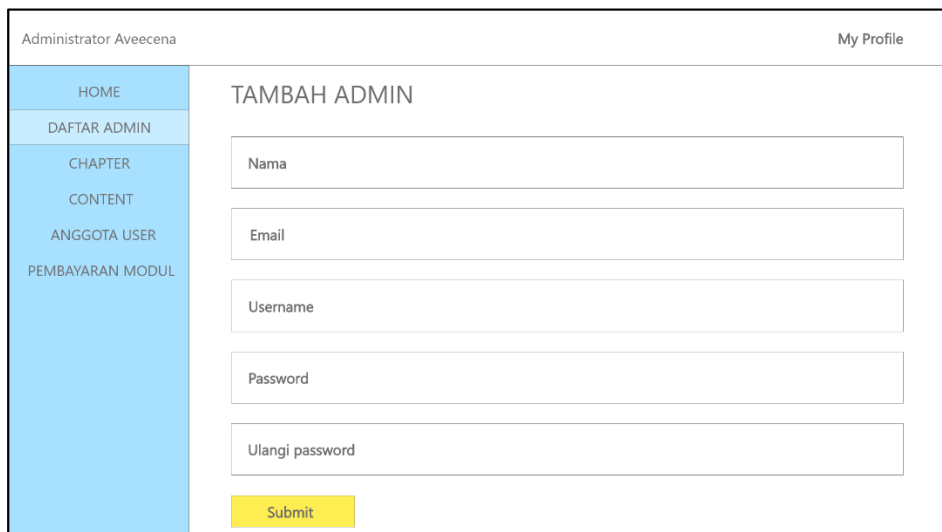
**Gambar 3. 7 UI *Home Admin***

Pada halaman *Home*, rencananya akan ditampilkan grafik-grafik statistik, seperti statistik total pengguna, total pemasukan, dan beberapa grafik lain yang nanti akan dibahas lebih lanjut.



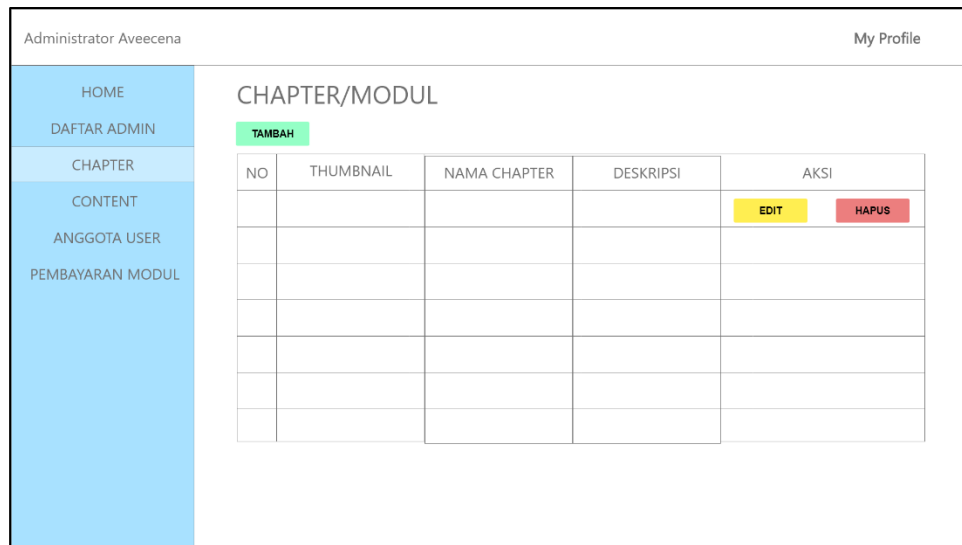
**Gambar 3. 8 UI Daftar Admin**

Pada halaman *Daftar Admin*, akan ditampilkan daftar admin dalam bentuk tabel. Tabel berisikan nama admin, *username*, serta *email*.



**Gambar 3. 9 UI Formulir Tambah Admin**

Untuk menambahkan admin, maka harus mengisi formulir yang terdiri dari *input field* nama, *email*, *username*, *password*, dan ulangi *password*.



**Gambar 3. 10 UI Daftar Chapter**

Pada halaman *Daftar Chapter*, akan ditampilkan daftar *chapter* yang sudah ditambahkan, dalam bentuk tabel. Untuk menambahkan atau membuat *chapter* baru, admin perlu klik tombol *Tambah Chapter* yang terletak di atas tabel. Setelah itu admin mengisi formulir yang tersedia. Jika *chapter* sudah berhasil dibuat, maka akan muncul dalam tabel tersebut. *Chapter* yang sudah dibuat bisa diubah datanya dengan melakukan klik pada tombol *Edit* di kolom *Aksi*, dan bisa dihapus dengan melakukan klik pada tombol *Hapus* di kolom *Aksi*.

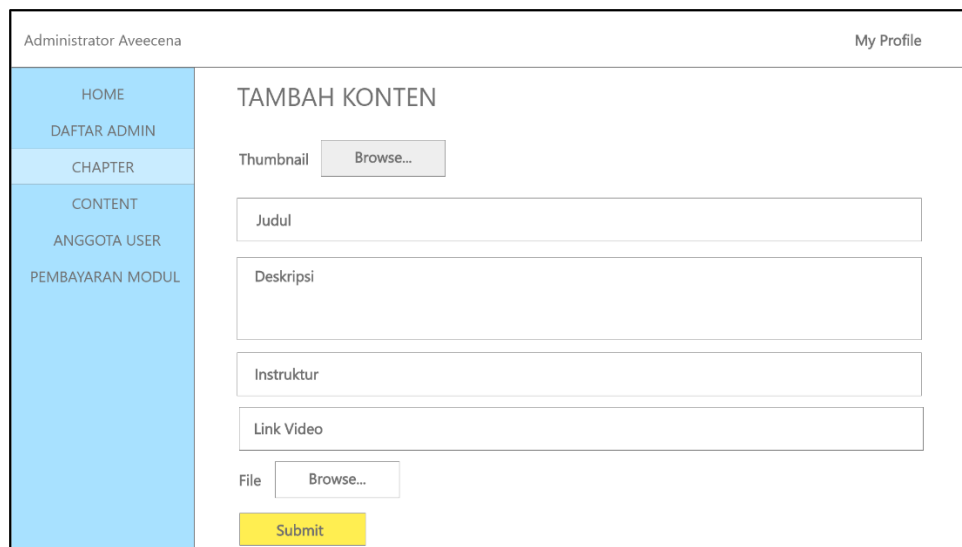
**Gambar 3. 11 UI Formulir Tambah *Chapter***

Untuk menambahkan *chapter*, maka harus mengisi formulir yang terdiri dari *input field* nama, deskripsi, aksesnya (apakah gratis atau berbayar). Jika akses berbayar, maka akan muncul *input field* untuk harga dalam mata uang rupiah.

NO	THUMBNAIL	JUDUL	DESKRIPSI	INSTRUKTUR	VIDEO	FILE	AKSI
							EDIT
							HAPUS

**Gambar 3. 12 UI Daftar Konten**

Pada halaman *Daftar Konten*, akan ditampilkan daftar konten yang sudah ditambahkan, dalam bentuk tabel. Untuk menambahkan atau membuat konten baru, admin perlu klik tombol *Tambah Konten* yang terletak di atas tabel. Setelah itu admin mengisi formulir yang tersedia. Jika konten sudah berhasil dibuat, maka akan muncul dalam tabel tersebut. Konten yang sudah dibuat bisa diubah datanya dengan melakukan klik pada tombol *Edit* di kolom *Aksi*, dan bisa dihapus dengan melakukan klik pada tombol *Hapus* di kolom *Aksi*.



The screenshot shows a web interface for an administrator. At the top left, it says 'Administrator Aveecena' and at the top right 'My Profile'. On the left side, there is a vertical navigation menu with the following items: HOME, DAFTAR ADMIN, CHAPTER, CONTENT (highlighted in blue), ANGGOTA USER, and PEMBAYARAN MODUL. The main content area is titled 'TAMBAH KONTEN' and contains the following form fields: a 'Thumbnail' field with a 'Browse...' button, a 'Judul' (Title) text input field, a 'Deskripsi' (Description) text area, an 'Instruktur' (Instructor) text input field, a 'Link Video' text input field, a 'File' field with a 'Browse...' button, and a yellow 'Submit' button at the bottom.

**Gambar 3. 13 Formulir Tambah Konten**

Untuk menambahkan konten, maka harus mengisi formulir yang terdiri dari *input field* unggah *thumbnail*, judul, deskripsi, instruktur, *link* video, dan unggah *file*.

Administrator Aveecena		My Profile	
HOME	DAFTAR ANGGOTA		
DAFTAR ADMIN			
CHAPTER			
CONTENT			
ANGGOTA USER			
PEMBAYARAN MODUL			
	NO	NAMA USER	USERNAME
			EMAIL

**Gambar 3. 14 Halaman Daftar Anggota**

Pada halaman *Daftar Anggota*, akan ditampilkan daftar anggota atau *user* yang sudah melakukan registrasi, dalam bentuk tabel.

Administrator Aveecena		My Profile		
HOME	PEMBAYARAN MODUL			
DAFTAR ADMIN				
CHAPTER				
CONTENT				
ANGGOTA USER				
PEMBAYARAN MODUL				
	NO	USERNAME	NAMA CHAPTER	TAGIHAN
				PELUNASAN

**Gambar 3. 15 Halaman Daftar Pembayaran**

Pada halaman *Pembayaran Modul*, akan ditampilkan daftar tagihan untuk *chapter* berbayar, baik itu yang sudah lunas maupun belum, dalam bentuk tabel.

### 3.2.5. Minggu Kelima: Mengembangkan *front-end dashboard* admin (*index*, *session*, koneksi *database*)

Setelah selesai membuat *interface* untuk *dashboard user* dan *dashboard admin*, pada minggu berikutnya mulai dilakukan pengembangan sistem, menggunakan *HTML5*, *CSS3*, *Bootstrap 4.4*, *JavaScript*, *PHP 7*, dan *MySQL*. Pengembangan dimulai dari membuat kode *master*, *index*, dan halaman *login* terlebih dahulu.

```
<nav class="navbar navbar-expand-lg navbar-light bg-white shadow rounded">
  <a class="navbar-brand font-weight-bolder" href="#" style="color: #1bb3eb; font-family: open sans;" >
    
    Administrator E-Course Aveecena
  </a>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
          data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          <i class="fas fa-user-circle" title="User"></i>
        </a>
        <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
          <a class="dropdown-item" href="index.php?page=profile">Profil</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="profil_proses.php?action=logout">Log Out</a>
        </div>
      </li>
    </ul>
  </div>
</nav>
```

**Gambar 3. 16 Kode Top Navbar**

Gambar 3.16 merupakan kode untuk *top navigation bar* yang berisikan logo Aveecena, dengan *navbar brand Administrator E-Course Aveecena*, dan *dropdown list* untuk membuka halaman *profile*.



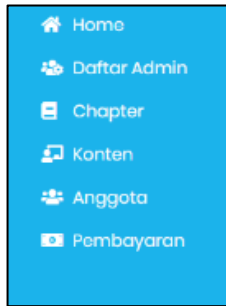
**Gambar 3. 17 Tampilan *Top Navbar***

Kode pada Gambar 3.16 menghasilkan tampilan *top navigation bar* seperti yang terlihat pada Gambar 3.17.

```
<nav class="col-md-2 sidebar rounded" style="min-height:100vh; background-color: #1bb3eb;" >
  <div class="sidebar-sticky">
    <ul class="nav flex-column">
      <li class="nav-item">
        <a class="nav-link text-white" href="index.php?page=home">
          <i class="fas fa-home" style="width: 25px;"></i> Home</a>
        </li>
      <li class="nav-item">
        <a class="nav-link text-white active" href="index.php?page=admin">
          <i class="fas fa-users-cog" style="width: 25px;"></i> Daftar Admin</a>
        </li>
      <li class="nav-item">
        <a class="nav-link text-white" href="index.php?page=chapter">
          <i class="fas fa-book" style="width: 25px;"></i> Chapter</a>
        </li>
      <li class="nav-item">
        <a class="nav-link text-white" href="index.php?page=content">
          <i class="fas fa-chalkboard-teacher" style="width: 25px;"></i> Konten</a>
        </li>
      <li class="nav-item">
        <a class="nav-link text-white" href="index.php?page=">
          <i class="fas fa-users" style="width: 25px;"></i> Anggota</a>
        </li>
      <li class="nav-item">
        <a class="nav-link text-white" href="#">
          <i class="fas fa-money-bill-alt" style="width: 25px;"></i> Pembayaran</a>
        </li>
    </ul>
  </div>
</nav>
```

**Gambar 3. 18 Kode *Side Navbar***

Gambar 3.18 merupakan kode untuk *side navigation bar* yang berisikan menu *Home*, *Daftar Admin*, *Chapter*, *Content*, *Anggota*, dan *Pembayaran*.



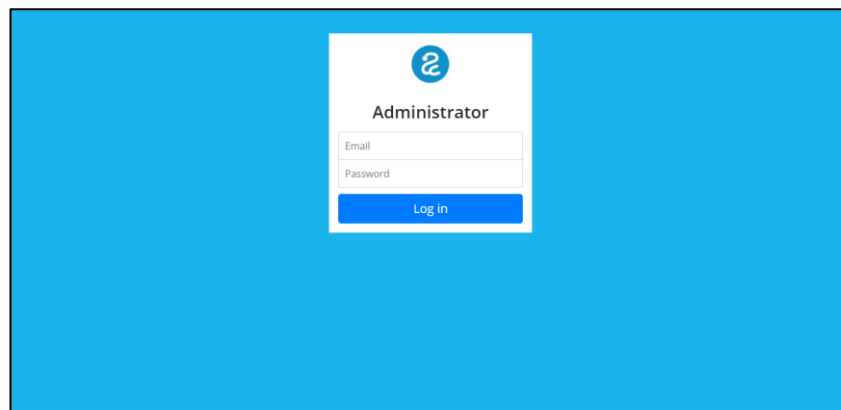
**Gambar 3. 19 Side Navbar**

Kode pada Gambar 3.18 menghasilkan tampilan *side navigation bar* seperti yang terlihat pada Gambar 3.19.

```
<form class="form-signin bg-white" method="POST" action="login_proses.php">
  
  <h1 class="h3 mb-3 font-weight-bolder">Administrator</h1>
  <input type="email" name="email" id="inputEmail" class="form-control" placeholder="Email " required autofocus>
  <input type="password" name="password" id="inputPassword" class="form-control" placeholder="Password" required>
  <button class="btn btn-lg btn-primary btn-block" type="submit">Log in</button>
</form>
```

**Gambar 3. 20 Kode Tampilan Login**

Gambar 3.20 merupakan kode untuk tampilan *login*, yang mana berisikan sebuah formulir *log in* dengan *input field Email* dan *Password*.



**Gambar 3. 21 Login**

Kode pada Gambar 3.20 menghasilkan tampilan *log in* seperti yang terlihat pada Gambar 3.21.

```
<?php
session_start();
if(empty($_SESSION['email']))
{
    echo "<script type='text/javascript'>
        alert('Anda harus log in terlebih dahulu. ');
        window.location='login.php'</script>";
    exit;
}
```

**Gambar 3. 22 Kode Session**

Gambar 3.22 merupakan kode untuk *session*, di mana apabila belum ada *session* yang terbentuk, maka pengguna diarahkan ke halaman *log in* untuk masuk ke akunnya terlebih dahulu.

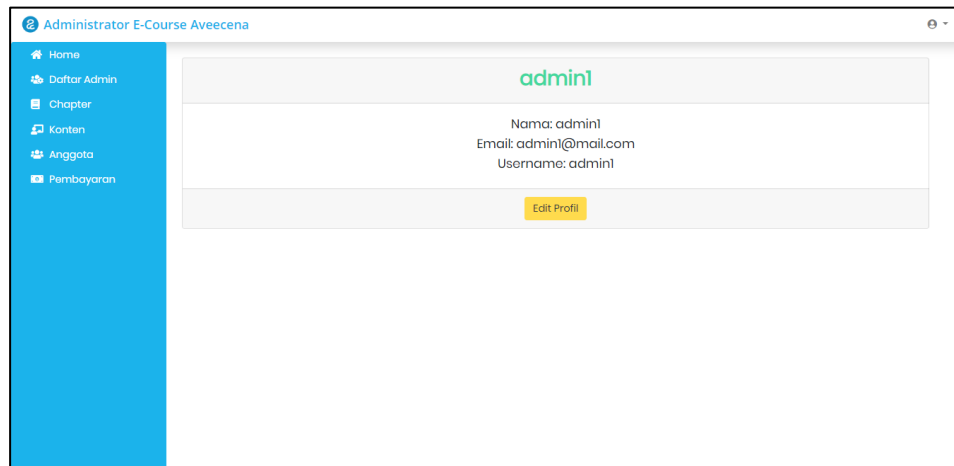
### 3.2.6. Minggu Keenam: Mengembangkan *front-end dashboard* admin (laman *profile*, daftar admin)

Pada minggu berikutnya, pengembangan dilakukan untuk halaman profil serta daftar admin.

```
<div class="card text-center">
<div class="card-header">
<h1 class="h2 font-weight-bolder text-center" style="color: #44d89e;"><?php echo $_SESSION['nama']; ?></h1>
</div>
<div class="card-body">
<h5>Nama: <?= $_SESSION['nama']; ?></h5>
<h5>Email: <?= $_SESSION['email']; ?></h5>
<h5>Username: <?= $_SESSION['username']; ?></h5>
</div>
<div class="card-footer text-muted">
<a href="index.php?page=profilform&id=<?= $_SESSION['id']; ?>"
class="btn text-dark" style="background-color: #ffdb4d;">Edit Profil</a>
</div>
</div>
```

**Gambar 3. 23 Kode Halaman Profile**

Gambar 3.23 merupakan kode untuk halaman *profile*, di mana di dalamnya berisikan informasi dasar profil admin tersebut seperti nama, *email*, serta *username*.



**Gambar 3. 24 Halaman *Profile***

Kode pada Gambar 3.23 menghasilkan tampilan profil seperti yang terlihat pada Gambar 3.24.

```
<h1 class="h2 font-weight-bolder text-center" style="color: #44d89e;">Profil</h1>
<form action="profil_proses.php?action=edit&id=<?>= &$_SESSION['id']; <?>" method="post">
  <div class="form-group">
    <label>Nama</label>
    <input type="text" class="form-control" name="nama" placeholder="Nama" value="<?>= $nama; <?>" required>
  </div>
  <div class="form-group">
    <label>Email</label>
    <input type="email" class="form-control" name="email" placeholder="email" value="<?>= $email; <?>" required>
  </div>
  <div class="form-group">
    <label>Username</label>
    <input type="text" class="form-control" name="username" placeholder="username" value="<?>= $username; <?>" required>
  </div>
  <div class="form-group">
    <label>Password</label>
    <input type="text" class="form-control" name="password" placeholder="password" value="<?>= $password; <?>" required>
  </div>
  <button type="submit" class="btn btn-warning">Submit Perubahan</button>
</form>
```

**Gambar 3. 25 Kode Halaman *Edit Profile***

Gambar 3.25 merupakan kode untuk halaman *edit profile*, yang mana bersisikan sebuah formulir untuk mengganti nama, *email*, *username*, ataupun *password*.

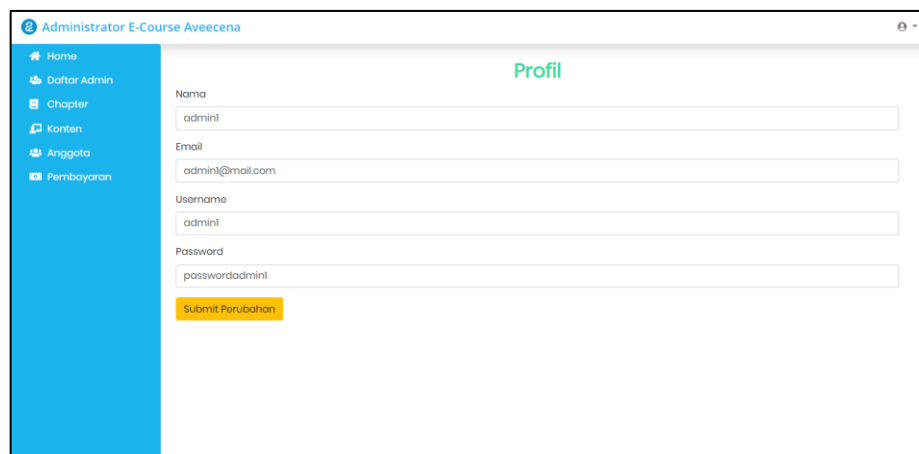
```
if($_GET['action'] == "edit"){
    $nama = $_POST['nama'];
    $email = $_POST['email'];
    $username = $_POST['username'];
    $password = $_POST['password'];

    $sql = "UPDATE admin
    SET nama = ?, email = ?, username = ?, password = ? WHERE id = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$nama, $email, $username, $password, $_GET['id']]);

    echo "<script type='text/javascript'>
        alert('Profil berhasil diedit.');
```

**Gambar 3. 26 Kode Proses *Edit Profile***

Gambar 3.26 merupakan kode proses *update* untuk membuat perubahan data profil.



The screenshot shows a web application interface for an administrator. On the left is a blue sidebar with navigation links: Home, Daftar Admin, Chapter, Konten, Anggota, and Pembayaran. The main content area is titled 'Profil' and contains a form with the following fields: 'Nama' (admin), 'Email' (admin@mail.com), 'Username' (admin), and 'Password' (passwordadmin). A yellow button labeled 'Submit Perubahan' is positioned below the password field.

**Gambar 3. 27 Tampilan Formulir *Profile***

Kode pada Gambar 3.26 menghasilkan tampilan profil seperti yang terlihat pada Gambar 3.27.

```
<?php
require 'db.php';

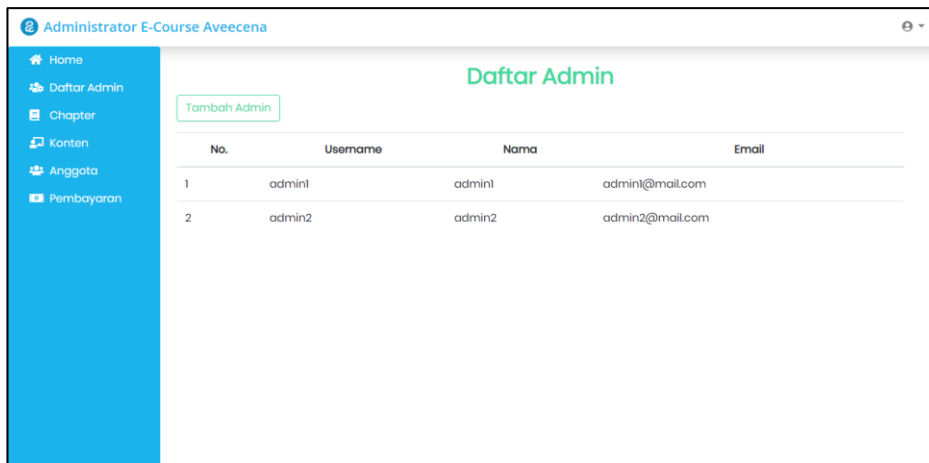
$sql = "SELECT * FROM admin";
$hasil = $pdo->prepare($sql);
$hasil->execute();
??

<h1 class="h2 font-weight-bolder text-center" style="color: #44d89e;">Daftar Admin</h1>
<a href="index.php?page=adminform"
class="btn btn-white shadow-sm" style="color: #44d89e; border-color: #44d89e;">Tambah Admin</a>

<div class="table-responsive-md mt-3">
  <table class="table table-hover">
    <thead class="text-center">
      <tr>
        <th>No.</th>
        <th>Username</th>
        <th>Nama</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <?php
      $i = 0;
      while($row = $hasil->fetch()):
      $i++;
      ?>
      <tr>
        <td><?= $i;></td>
        <td><?= $row['username'];></td>
        <td><?= $row['nama'];></td>
        <td><?= $row['email'];></td>
      </tr>
      <?php endwhile; ?>
    </tbody>
  </table>
</div>
```

**Gambar 3. 28 Kode Halaman Daftar Admin**

Gambar 3.28 merupakan kode untuk halaman daftar admin. Pada halaman tersebut, daftar admin ditampilkan dalam sebuah tabel.



**Gambar 3. 29 Tampilan Halaman Daftar Admin**

Kode pada Gambar 3.28 menghasilkan tampilan halaman daftar admin seperti yang terlihat pada Gambar 3.29.

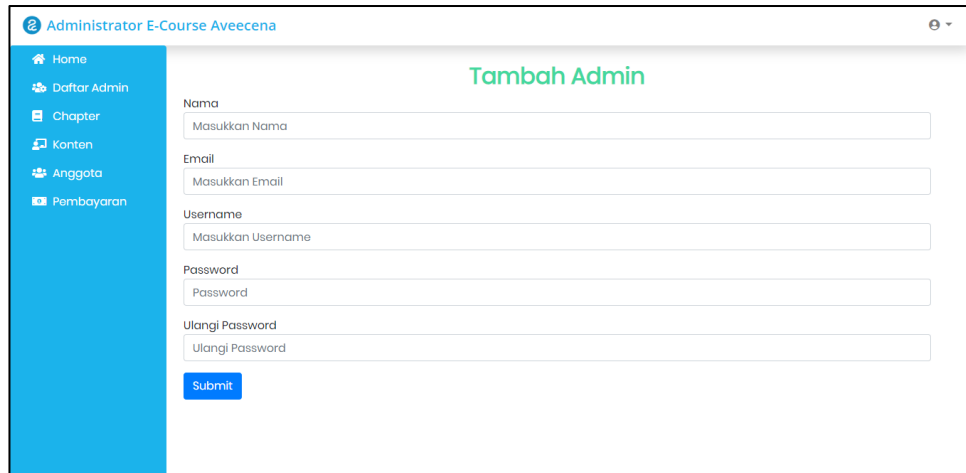
```

<h1 class="h2 font-weight-bolder text-center" style="color: #44d89e;">Tambah Admin</h1>
<form action="admin_proses.php?action=tambah" method="post">
<div class="form-group">
  Nama
  <input type="text" class="form-control" name="nama" placeholder="Masukkan Nama" required>
</div>
<div class="form-group">
  Email
  <input type="email" class="form-control" name="email" placeholder="Masukkan Email" required>
</div>
<div class="form-group">
  Username
  <input type="text" class="form-control" name="username" placeholder="Masukkan Username" required>
</div>
<div class="form-group">
  Password
  <input type="password" class="form-control" name="password" placeholder="Password" required>
</div>
<div class="form-group">
  Ulangi Password
  <input type="password" class="form-control" name="ulangipassword" placeholder="Ulangi Password" required>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>

```

**Gambar 3. 30 Kode Halaman Tambah Admin**

Gambar 3.30 merupakan kode untuk halaman tambah admin. Halaman tersebut berisikan sebuah formulir dengan *input field* nama, *email*, *username*, *password*, dan *ulangi password*.



The screenshot shows a web application interface for adding an administrator. The title bar reads 'Administrator E-Course Aveecena'. A blue sidebar on the left contains the following menu items: Home, Daftar Admin, Chapter, Konten, Anggota, and Pembayaran. The main content area is titled 'Tambah Admin' in green text. Below the title, there is a form with five input fields: 'Nama' (Masukkan Nama), 'Email' (Masukkan Email), 'Username' (Masukkan Username), 'Password' (Password), and 'Ulangi Password' (Ulangi Password). A blue 'Submit' button is positioned at the bottom of the form.

**Gambar 3. 31 Tampilan Halaman Tambah Admin**

Kode pada Gambar 3.30 menghasilkan tampilan halaman daftar admin seperti yang terlihat pada Gambar 3.31.



```

$sql1 = "SELECT email FROM admin where email='$email'";
$stmt1 = $pdo->prepare($sql1);
$stmt1->execute();
$sql2 = "SELECT username FROM admin where username='$username'";
$stmt2 = $pdo->prepare($sql2);
$stmt2->execute();
if($stmt1->fetch()){
    echo "<script type='text/javascript'>
        alert('Sudah ada akun admin dengan email tersebut. Mohon gunakan email lain.');"
        window.location = 'index.php?page=adminform';
    </script>";
}elseif($stmt2->fetch()){
    echo "<script type='text/javascript'>
        alert('Sudah ada akun admin dengan username tersebut. Mohon gunakan username lain.');"
        window.location = 'index.php?page=adminform';
    </script>";
} else {
    if($password == $ulangipassword){
        $sql = "INSERT INTO admin (nama, email, username, password)
        VALUES (?, ?, ?, ?)";
        $hasil = $pdo->prepare($sql);
        $hasil->execute([$nama, $email, $username, $password]);
        if($hasil->rowCount() > 0){
            echo "<script type='text/javascript'>
                alert('Admin berhasil ditambahkan.');"
                window.location = 'index.php?page=admin';
            </script>";
        }
    }elseif($password != $ulangipassword){
        echo "<script type='text/javascript'>
            alert('Password tidak sama.');"
            window.location = 'index.php?page=adminform';
        </script>";
    }
}
}

```

**Gambar 3. 32 Kode Proses Tambah Admin**

Gambar 3.32 merupakan kode proses *create* untuk membuat data admin baru ke dalam *database*.

### 3.2.7. Minggu Ketujuh: Mengembangkan *front-end dashboard* admin (laman *chapter*)

Pada minggu berikutnya, dilakukan pengembangan pada halaman *chapter*. Halaman ini merupakan halaman untuk membuat *chapter* baru serta melihat daftar *chapter* yang sudah dibuat.

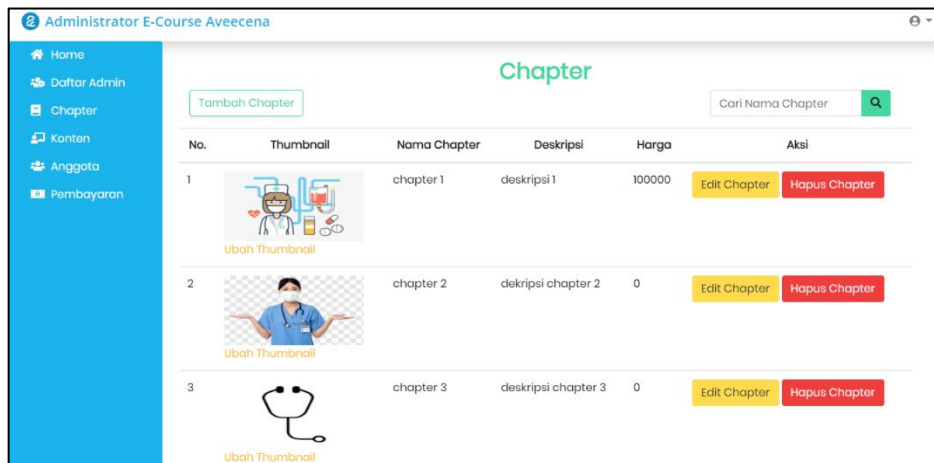
```

<table class="table table-hover" id="mytable">
  <thead class="text-center">
    <tr>
      <th>No.</th>
      <th>Thumbnail</th>
      <th>Nama Chapter</th>
      <th>Deskripsi</th>
      <th>Harga</th>
      <th>Aksi</th>
    </tr>
  </thead>
  <tbody>
    <?php
    $i = 0;
    while($row = $hasil->fetch()):
    $i++;
    ?>
    <tr>
      <td><? = $i;?></td>
      <td><br/>
      <a href="index.php?page=chapterform&action=editthumbnail&id=<? = $row['id']; ?>"
      class="text-warning">Ubah Thumbnail</a></td>
      <td><? = $row['nama'];?></td>
      <td><? = $row['deskripsi'];?></td>
      <td><? = $row['harga'];?></td>
      <td>
        <a href="index.php?page=chapterform&action=edit&id=<? = $row['id']; ?>"
        class="btn text-dark" style="background-color: #ffdb4d;">Edit Chapter</a>
        <a href="chapter_proses.php?action=delete&id=<? = $row['id']; ?>"
        class="btn text-light" style="background-color: #f03e3e;">Hapus Chapter</a>
      </td>
    </tr>
    <?php endwhile; ?>
  </tbody>
</table>

```

**Gambar 3. 33 Kode Halaman Chapter**

Gambar 3.33 merupakan kode untuk halaman *chapter*. Pada halaman tersebut, daftar *chapter* ditampilkan dalam sebuah tabel. Terdapat pula *button* untuk mengarahkan ke halaman tambah *chapter*, *edit chapter*, atau hapus *chapter*. Terdapat pula sebuah *search bar* untuk melakukan pencarian berdasarkan nama *chapter*.



**Gambar 3. 34 Tampilan Halaman *Chapter***

Kode pada Gambar 3.34 menghasilkan tampilan halaman *chapter* seperti pada Gambar 3.34.

```

<script>
function search() {
  var input, filter, table, tr, td, i, txtValue;
  input = document.getElementById("katapencarian");
  filter = input.value.toUpperCase();
  table = document.getElementById("mytable");
  tr = table.getElementsByTagName("tr");

  for (i = 0; i < tr.length; i++) {
    td = tr[i].getElementsByTagName("td")[2];
    if (td) {
      txtValue = td.textContent || td.innerText;
      if (txtValue.toUpperCase().indexOf(filter) > -1) {
        tr[i].style.display = "";
      } else {
        tr[i].style.display = "none";
      }
    }
  }
}
</script>

```

**Gambar 3. 35 Kode Proses *Search Chapter***

Gambar 3.35 merupakan kode *Javascript* untuk melakukan pencarian nama *chapter* dari *search bar*.

```

<form action="chapter_proses.php?action=tambah" method="POST" enctype="multipart/form-data">
<div class="form-group" id="inputthumbnail" >
<label>Thumbnail</label><br>
<input type="file" name="foto" value="<?=$thumbnail; ?>" required><br>
</div>
<div class="form-group">
<label>Nama</label>
<input type="text" class="form-control" name="nama" placeholder="Nama Chapter" value="<?=$nama; ?>" required>
</div>
<div class="form-group">
<label>Deskripsi</label>
<textarea class="form-control" rows="5" name="deskripsi" placeholder="Deskripsi Chapter" required><?=$deskripsi; ?></textarea>
</div>
<div class="form-group">
<label>Akses:</label>
<div class="checkbox">
<input type="checkbox" name="hargagratiss" value="0">Gratis</label>
</div>
<div class="checkbox">
<input type="checkbox" value="" id="checkboxbayar" onclick="berbayar()">Berbayar</label>
</div>
<div class="form-group" style="display:none" id="harga">
<label>Harga (Rp)</label>
<input type="text" class="form-control" name="hargabayar" value="<?=$harga; ?>" placeholder="Harga Chapter" required>
</div>
<button type="submit" class="btn btn-warning">Submit</button>
</form>

```

**Gambar 3. 36 Kode Formulir *Chapter***

Gambar 3.36 merupakan kode untuk formulir *chapter*. Pada halaman tersebut, terdapat sebuah formulir yang berisi *input field upload thumbnail*, nama, deskripsi, akses, dan harga *chapter*. Jika admin klik tombol tambah *chapter*, maka *input field* ditampilkan tanpa *value*. Jika admin klik tombol *edit chapter*, maka *input field* akan menampilkan *value* berdasarkan *database chapter* tersebut.

**Gambar 3. 37 Tampilan Formulir *Chapter***

Kode pada Gambar 3.36 menghasilkan tampilan formulir *chapter* seperti pada Gambar 3.37.

```
if($_GET['action'] == "tambah"){
    $nama = $_POST['nama'];
    $deskripsi = $_POST['deskripsi'];

    if(isset($_POST['hargagratis'])){
        $harga = $_POST['hargagratis'];
    } elseif(isset($_POST['hargabayar'])){
        $harga = $_POST['hargabayar'];
    }

    $sql1 = "SELECT nama FROM chapter where nama='$nama'";
    $stmt1 = $pdo->prepare($sql1);
    $stmt1->execute();

    if($stmt1->fetch()){
        echo "<script type='text/javascript'>
        alert('Sudah ada chapter dengan nama ini. Mohon gunakan nama lain.');"
        window.location = 'index.php?page=tambahchapter';
        </script>";
    } else {
        $ext = explode(".", $_FILES['foto']['name']);
        $ext = end($ext);
        $ext = strtolower($ext);
        $ext_boleh = ["jpg", "jpeg", "png"];
        if(in_array($ext, $ext_boleh)){
            $tmp = $_FILES['foto']['tmp_name'];
            $permanent = "pic/" . $_FILES['foto']['name'];
            move_uploaded_file($tmp, $permanent);

            $sql = "INSERT INTO chapter (thumbnail, nama, deskripsi, harga)
            VALUES (?, ?, ?, ?)";
            $hasil = $pdo->prepare($sql);
            $hasil->execute([$permanent, $nama, $deskripsi, $harga]);
            echo "<script type='text/javascript'>
            alert('Chapter berhasil ditambahkan.');"
            window.location = 'index.php?page=chapter';
            </script>";
        }
    }
}
```

**Gambar 3. 38 Kode Proses Tambah Chapter**

Gambar 3.38 merupakan kode proses *create* untuk membuat data *chapter* baru ke dalam *database*.

```

} elseif($_GET['action'] == "edit"){
    $nama = $_POST['nama'];
    $deskripsi = $_POST['deskripsi'];

    if(isset($_POST['hargagratis'])){
        $harga = $_POST['hargagratis'];
    } elseif(isset($_POST['hargabayar'])){
        $harga = $_POST['hargabayar'];
    }

    $sql = "UPDATE chapter SET nama = ?, deskripsi = ?, harga = ? WHERE id = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$nama, $deskripsi, $harga, $_GET['id']]);

    echo "<script type='text/javascript'>
    alert('Chapter berhasil diedit.');

```

**Gambar 3. 39 Kode Proses Edit Chapter**

Gambar 3.39 merupakan kode proses *update* untuk membuat perubahan data *chapter* yang sudah ada dengan data yang baru ke dalam *database*.

```

} elseif($_GET['action'] == "editthumbnail"){
    $ext = explode(".", $_FILES['foto']['name']);
    $ext = end($ext);
    $ext = strtolower($ext);
    $ext_boleh = ["jpg", "jpeg", "png"];
    if(in_array($ext, $ext_boleh)){
        $tmp = $_FILES['foto']['tmp_name'];
        $permanent = "pic/" . $_FILES['foto']['name'];
        move_uploaded_file($tmp, $permanent);

        $sql = "UPDATE chapter SET thumbnail = ? WHERE id = ?";
        $stmt = $pdo->prepare($sql);
        $stmt->execute([$permanent, $_GET['id']]);

        echo "<script type='text/javascript'>
        alert('Thumbnail chapter berhasil diubah.');

```

**Gambar 3. 40 Kode Proses Edit Thumbnail Chapter**

Gambar 3.40 merupakan kode proses *update* untuk membuat perubahan *thumbnail chapter* yang sudah ada dengan *file thumbnail* baru ke dalam *database*.

```
    } elseif($_GET['action'] == "delete"){  
        $sql = "DELETE FROM chapter WHERE id = ?";  
        $hasil = $pdo->prepare($sql);  
        $hasil->execute([$GET['id']]);  
        echo "<script type='text/javascript'>  
        alert('Chapter berhasil dihapus.');        window.location = 'index.php?page=chapter';  
        </script>";  
    }
```

**Gambar 3. 41 Kode Proses *Delete Chapter***

Gambar 3.41 merupakan kode proses *delete* untuk menghapus data *chapter* yang ada pada *database*.

### 3.2.8. Minggu Kedelapan: Mengembangkan *front-end dashboard* admin (laman *content*)

Pada minggu berikutnya, dilakukan pengembangan pada halaman *content*. Halaman ini merupakan halaman untuk membuat *content* baru serta melihat daftar *content* yang sudah dibuat.

```

<table class="table table-hover" id="mytable">
  <thead class="text-center">
    <tr>
      <th>No.</th> <th>Thumbnail</th> <th>Judul Konten</th>
      <th>Deskripsi</th> <th>Chapter</th> <th>Instruktur</th>
      <th>Video</th> <th>File</th> <th>Aksi</th> </tr>
    </thead>
    <tbody>
      <?php
      $i = 0;
      while($row = $hasil->fetch()):
      $i++; ?>
      <tr>
        <td>?<= $i;?</td>
        <td><br>
        <a href="index.php?page=contentform&action=editthumbnail&id=<?<= $row['id']; ?>"
        class="text-warning">Ubah Thumbnail</a></td>
        <td>?<= $row['judul'];?</td> <td>?<= $row['deskripsi'];?</td>
        <td>?<= $row['chapter'];?</td> <td>?<= $row['instruktur'];?</td>
        <td>
          <?php
          $url=$row['video'];
          $fetch=explode("v=", $url);
          $videoid=$fetch[1];
          echo '<object width="200" data="http://www.youtube.com/v/'. $videoid.'"
          type="application/x-shockwave-flash"><param name="src" value="http://www.youtube.com/v/'. $videoid.'" />
          </object>' ?>
        </td>
        <td><a href="<?<= $row['filekonten'];?>" target="_blank">Download File</a><br>
        <a href="index.php?page=contentform&action=editfile&id=<?<= $row['id']; ?>"
        class="text-warning">Ubah File</a></td>
        <td>
          <a href="index.php?page=contentform&action=edit&id=<?<= $row['id']; ?>"
          class="btn text-dark" style="background-color: #ffdb4d;">Edit Konten</a>
          <a href="content_proses.php?action=delete&id=<?<= $row['id']; ?>"
          class="btn text-light" style="background-color: #f03e3e;">Hapus Konten</a>
        </td>
      </tr>
      <?php endwhile; ?>
    </tbody>
  </table>

```

**Gambar 3. 42 Kode Halaman Konten**

Gambar 3.33 merupakan kode untuk halaman konten. Pada halaman tersebut, daftar konten ditampilkan dalam sebuah tabel. Terdapat pula *button* untuk mengarahkan ke halaman tambah konten, *edit* konten, atau hapus konten. Terdapat pula sebuah *search bar* untuk melakukan pencarian berdasarkan judul konten.



```

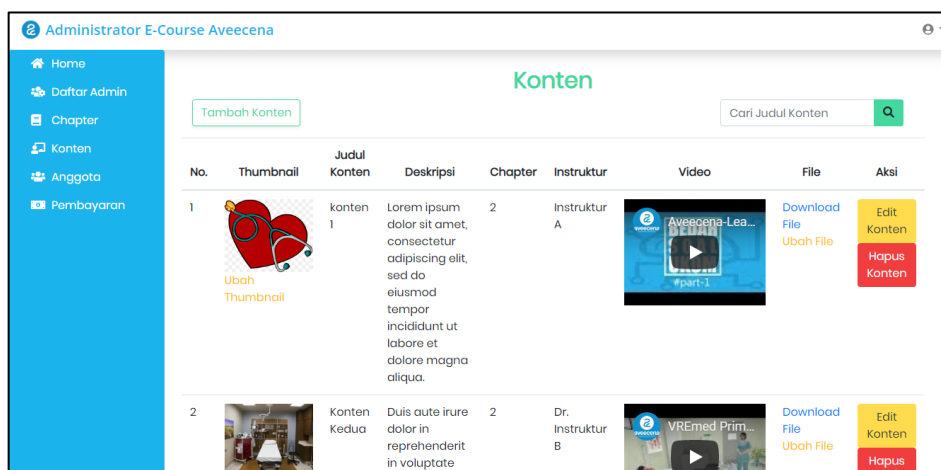
<script>
function search() {
var input, filter, table, tr, td, i, txtValue;
input = document.getElementById("katapencarian");
filter = input.value.toUpperCase();
table = document.getElementById("mytable");
tr = table.getElementsByTagName("tr");

for (i = 0; i < tr.length; i++) {
td = tr[i].getElementsByTagName("td")[2];
if (td) {
txtValue = td.textContent || td.innerHTML;
if (txtValue.toUpperCase().indexOf(filter) > -1) {
tr[i].style.display = "";
} else {
tr[i].style.display = "none";
}
}
}
}
</script>

```

**Gambar 3. 43 Kode Proses *Search Content***

Gambar 3.43 merupakan kode *Javascript* untuk melakukan pencarian judul konten dari *search bar*.



**Gambar 3. 44 Tampilan Halaman *Konten***

Pada Gambar 3.44, terlihat hasil jadi tampilan halaman *Konten*. Pada halaman tersebut ditampilkan sebuah tabel yang berisikan daftar konten dengan rincian data *thumbnail* konten, judul konten, deskripsi konten,

*chapter*, instruktur atau pengajar konten, video konten, serta *link* untuk mengunduh *file* materi konten.

```
<form action="content_proses.php?action=tambah" method="post" enctype="multipart/form-data">
<div class="form-group">
<label>Thumbnail</label><br>
<input type="file" name="thumbnail" value="<?=$thumbnail; ?>" required><br>
</div>
<div class="form-group">
<label>Judul</label>
<input type="text" class="form-control" name="judul" placeholder="Judul Konten" value="<?=$judul; ?>" required>
</div>
<div class="form-group">
<label>Deskripsi</label>
<textarea class="form-control" rows="5" name="deskripsi" placeholder="Deskripsi Konten" required><?=$deskripsi; ?></textarea>
</div>
<div class="form-group">
<label>Instruktur</label>
<input type="text" class="form-control" name="instruktur" placeholder="Instruktur" value="<?=$instruktur; ?>" required>
</div>
<div class="form-group">
<label>Link Video</label>
<input type="text" class="form-control" name="video" placeholder="Link Video" value="<?=$video; ?>" required>
</div>
<div class="form-group">
<label>File Konten</label><br>
<input type="file" name="filekonten" value="<?=$filekonten; ?>" required><br>
</div>
<div class="form-group">
<label>Chapter</label>
<select class="form-control" name="chapter">
<?php
while($row1 = $hasil->fetch()):
?>
<option value="<?=$row1['id']; ?>" <?=$row1['nama']; ?>" </option>
<?php endwhile; ?>
</select>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

**Gambar 3. 45 Kode Formulir Konten**

Gambar 3.45 merupakan kode untuk formulir konten. Pada halaman tersebut, terdapat sebuah formulir yang berisi *input field upload thumbnail*, judul, deskripsi, instruktur, *link video*, *upload file*, dan pilihan *chapter*. Jika admin klik tombol tambah konten, maka *input field* ditampilkan tanpa *value*. Jika admin klik tombol *edit konten*, maka *input field* akan menampilkan *value* berdasarkan *database* konten tersebut.

The image shows a web application interface for an e-course administrator. The page title is "Administrator E-Course Aveecena". On the left, there is a blue sidebar menu with the following items: Home, Daftar Admin, Chapter, Konten, Anggota, and Pembayaran. The main content area is titled "Content" and contains a form with the following fields:

- Thumbnail: Choose File | No file chosen
- Judul: Judul Konten
- Deskripsi: Deskripsi Konten
- Instruktur: Instruktur
- Link Video: Link Video
- File Konten: Choose File | No file chosen
- Chapter: chapter 1

A blue "Submit" button is located at the bottom of the form.

**Gambar 3. 46 Tampilan Formulir Konten**

Kode pada Gambar 3.36 menghasilkan tampilan formulir *chapter* seperti pada Gambar 3.37.

```

1
if($_GET['action'] == "tambah"){
    $judul = $_POST['judul'];
    $deskripsi = $_POST['deskripsi'];
    $instruktur = $_POST['instruktur'];
    $video = $_POST['video'];
    $chapter = $_POST['chapter'];

    $sql1 = "SELECT judul FROM content where judul='$judul'";
    $stmt1 = $pdo->prepare($sql1);
    $stmt1->execute();

    if($stmt1->fetch()){
        echo "<script type='text/javascript'>
        alert('Sudah ada konten dengan nama ini. Mohon gunakan nama lain atau buat konten lain. ');
        window.location = 'index.php?page=tambahchapter';
        </script>";
    } else {
        $ext = explode(".", $_FILES['thumbnail']['name']);
        $ext = end($ext);
        $ext = strtolower($ext);
        $ext_boleh = ["jpg","jpeg","png"];

        $ext2 = explode(".", $_FILES['filekonten']['name']);
        $ext2 = end($ext2);
        $ext2 = strtolower($ext2);
        $ext_boleh2 = ["doc","docx", "ppt", "pptx", "xls", "xlsx", "pdf"];

        if(in_array($ext, $ext_boleh)){
            $tmp = $_FILES['thumbnail']['tmp_name'];
            $permanent = "pic/" . $_FILES['thumbnail']['name'];
            move_uploaded_file($tmp, $permanent);

            if(in_array($ext2, $ext_boleh2)){
                $tmp2 = $_FILES['filekonten']['tmp_name'];
                $permanent2 = "filekonten/" . $_FILES['filekonten']['name'];
                move_uploaded_file($tmp2, $permanent2);

                $sql = "INSERT INTO content (thumbnail, judul, deskripsi, instruktur, video, filekonten, chapter)
                VALUES (?, ?, ?, ?, ?, ?, ?)";
                $hasil = $pdo->prepare($sql);
                $hasil->execute([$permanent, $judul, $deskripsi, $instruktur, $video, $permanent2, $chapter]);
                echo "<script type='text/javascript'>
                alert('Konten berhasil ditambahkan. ');
                window.location = 'index.php?page=content';
                </script>";
            }
        }
    }
}
}

```

**Gambar 3. 47 Kode Proses Tambah Konten**

Gambar 3.47 merupakan kode proses *create* untuk membuat data konten baru ke dalam *database*.

```

) elseif($_GET['action'] == "edit"){
    $judul = $_POST['judul'];
    $deskripsi = $_POST['deskripsi'];
    $instruktur = $_POST['instruktur'];
    $video = $_POST['video'];

    $sql1 = "UPDATE content SET judul = ?, deskripsi = ?, instruktur = ?, video = ? WHERE id = ?";
    $stmt1 = $pdo->prepare($sql1);
    $stmt1->execute([$judul, $deskripsi, $instruktur, $video, $_GET['id']]);

    echo "<script type='text/javascript'>
    alert('Konten berhasil diedit. ');
    window.location = 'index.php?page=content';
    </script>";
}

```

**Gambar 3. 48 Kode Proses Edit Konten**

Gambar 3.48 merupakan kode proses *update* untuk membuat perubahan data *chapter* yang sudah ada dengan data yang baru ke dalam *database*.

```
    } elseif($_GET['action'] == "editthumbnail"){
        $ext = explode(".", $_FILES['thumbnail']['name']);
        $ext = end($ext);
        $ext = strtolower($ext);
        $ext_boleh = ["jpg", "jpeg", "png"];
        if(in_array($ext, $ext_boleh)){
            $tmp = $_FILES['thumbnail']['tmp_name'];
            $permanent = "pic/" . $_FILES['thumbnail']['name'];
            move_uploaded_file($tmp, $permanent);

            $sql = "UPDATE content SET thumbnail = ? WHERE id = ?" ;
            $stmt = $pdo->prepare($sql);
            $stmt->execute([$permanent, $_GET['id']]);

            echo "<script type='text/javascript'>
                alert('Thumbnail konten berhasil diubah.');
```

**Gambar 3. 49 Kode Proses Edit *Thumbnail* Konten**

Gambar 3.49 merupakan kode proses *update* untuk membuat perubahan *thumbnail* konten yang sudah ada dengan *file thumbnail* baru ke dalam *database*.

```

} elseif($_GET['action'] == "editfile"){
    $ext2 = explode(".", $_FILES['filekonten']['name']);
    $ext2 = end($ext2);
    $ext2 = strtolower($ext2);
    $ext_boleh2 = ["doc", "docx", "ppt", "pptx", "xls", "xlsx", "pdf"];

    if(in_array($ext2, $ext_boleh2)){
        $tmp2 = $_FILES['filekonten']['tmp_name'];
        $permanent2 = "filekonten/" . $_FILES['filekonten']['name'];
        move_uploaded_file($tmp2, $permanent2);

        $sql = "UPDATE content SET filekonten = ? WHERE id = ?";
        $hasil = $pdo->prepare($sql);
        $hasil->execute([$permanent2, $_GET['id']]);
        echo "<script type='text/javascript'>
        alert('File konten berhasil diubah.');

```

**Gambar 3. 50 Kode Proses *Edit File Konten***

Gambar 3.50 merupakan kode proses *update* untuk membuat perubahan *file* konten yang sudah ada dengan *file* baru ke dalam *database*.

```

} elseif($_GET['action'] == "delete"){
    $sql = "DELETE FROM content WHERE id = ?";
    $hasil = $pdo->prepare($sql);
    $hasil->execute([$GET['id']]);

    echo "<script type='text/javascript'>
    alert('Konten berhasil dihapus.');

```

**Gambar 3. 51 Kode Proses *Delete Content***

Gambar 3.51 merupakan kode proses *delete* untuk menghapus data konten yang ada pada *database*.

### **3.3. Kendala yang Dihadapi**

Selama pelaksanaan magang sebagai *web developer* di Aveecena, ada beberapa kendala yang dihadapi, yaitu:

1. Pembimbing lapangan bukan berprofesi sebagai *web developer* ataupun berasal dari bidang terkait sistem atau teknologi informasi. Selain itu, tim pengembang hanya berisi sesama *intern*.
2. Adanya kesulitan untuk meminta tanda tangan berkas kepada pembimbing lapangan, karena pembimbing berprofesi sebagai perawat, sehingga sejak pertengahan bulan Maret beliau memiliki fokus lain yaitu untuk menjalankan tugas sebagai perawat pasien COVID-19 di rumah sakit dan menyebabkan keterlambatan dalam merespon pesan dan memproses dokumen.

### **3.4. Solusi atas Kendala**

Solusi yang dilakukan untuk mengatasi kendala-kendala yang dihadapi selama magang adalah:

1. Karena proyek dikerjakan bersama tim, meskipun tidak ada pembimbing lapangan yang paham secara teknis, anggota tim masih bisa saling bertanya dan berdiskusi antara satu sama lain. Informasi teknis juga dapat dicari dengan melakukan *browsing* atau mencari *tutorial* di Internet.
2. Dokumen-dokumen yang membutuhkan tanda tangan diberikan semuanya dalam waktu bersamaan, agar dapat diproses sekaligus. Jika dalam beberapa hari atau minggu belum mendapatkan respon, maka

kendala diatasi dengan cara mengirimkan pesan kembali melalui *WhatsApp* sebagai pengingat.