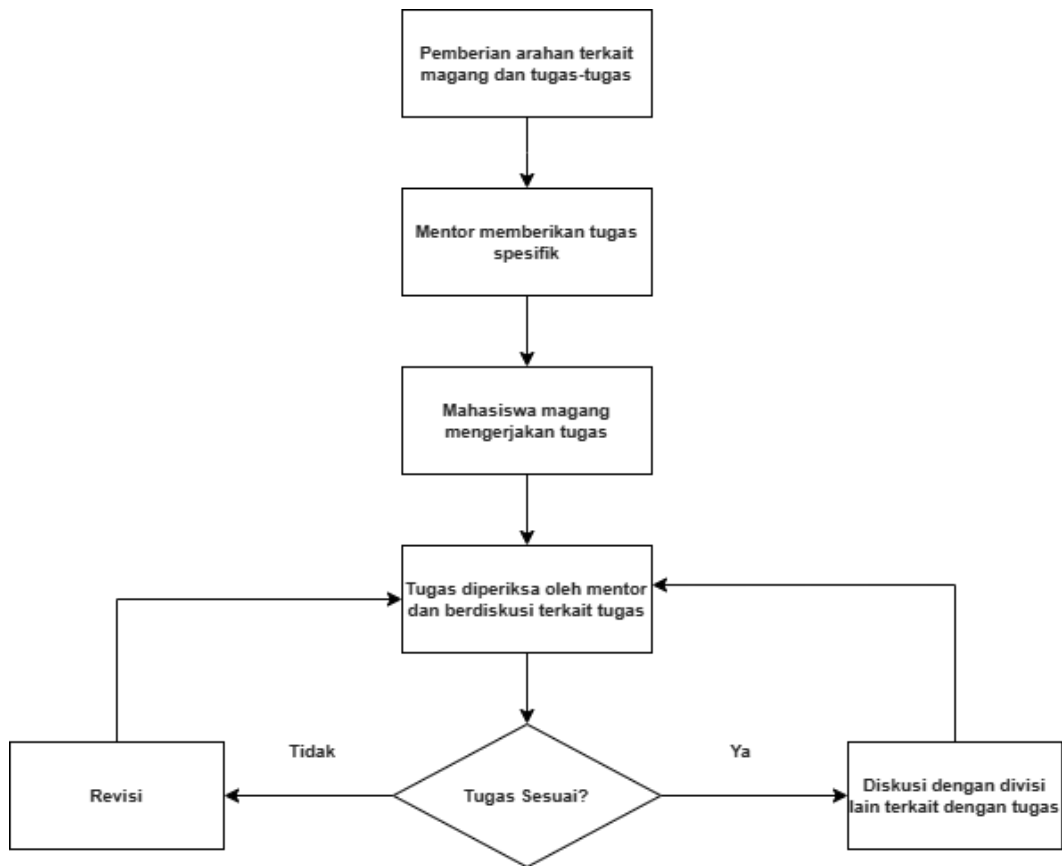


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Mahasiswa melaksanakan kerja magang di PT Nagatama Poin Ciaprima ditempatkan didepartemen *Full Stack Developer*. Mahasiswa berposisi sebagai *internship full stack developer* yang salah satu fokus utamanya adalah pada *Front-end Development*, yang melibatkan pembuatan antarmuka pengguna yang interaktif dan responsif menggunakan *HTML*, *CSS*, *JavaScript*, dan *framework* seperti *React* yang nantinya akan menjadi sebuah *website* bagi *e-commerce* untuk PT. Nagatama Poin Ciaprima. Dalam koordinasi kerja, mahasiswa yang berada di divisi *Full Stack Developer* harus bekerja sama erat dengan tim lainnya, termasuk *Back-end Developers*, dan *Designers*. Adapun alur kerja dari *Full Stack Developer* pada PT. Nagatama Poin Ciaprima sebagai berikut :



Gambar 3. 1 Alur kerja Mahasiswa Magang

Gambar 3.1 menggambarkan jalur kerja yang diikuti oleh mahasiswa selama program magang mereka. Jalur ini mencakup serangkaian langkah dan aktivitas yang dirancang untuk membantu mahasiswa memahami dan menavigasi proses magang dengan efektif. Berikut merupakan penjelasan mengenai gambar 3.1 diatas:

1. Mahasiswa magang yang melakukan magang di PT. Niagatama Poin Ciaprima dalam divisi *IT Departemen*, mendapatkan pengenalan terhadap konsep dasar pemrograman seperti *HTML*, *CSS*, dan *JavaScript*. Selain itu, juga diberikan pemahaman mengenai peran dan tanggung jawab dalam *IT Departemen*, termasuk peran sebagai *full stack developer*

2. Setelah tugas diterima dan tanggal akhir ditentukan, mahasiswa magang dapat memulai pekerjaan atas tugas tersebut.
3. Jika selama proses pekerjaan terjadi hambatan atau kesulitan, mahasiswa magang dapat berbincang dengan rekan-rekan magang lainnya serta dengan mentor.
4. Setelah tugas selesai dan memenuhi semua persyaratan, mahasiswa magang akan mengadakan diskusi atau pembahasan dengan mentor serta divisi relevan untuk mendapatkan *feedback* dan pemahaman yang lebih mendalam tentang proyek tersebut.
5. Jika hasil kerja mahasiswa magang tidak memenuhi kriteria yang ditetapkan, mentor akan memberikan saran untuk melakukan perbaikan atau revisi. Setelah perbaikan atau revisi selesai, mentor akan melakukan pengecekan ulang untuk memastikan bahwa tugas tersebut telah memenuhi standar yang telah ditetapkan.

3.2 Tugas dan Uraian Kerja Magang

Tugas yang dilakukan pada magang di PT. Nagatama Poin Ciaprima adalah menjadi seorang *front-end* yang mengembangkan antarmuka pengguna (UI) untuk situs *website* dengan menerjemahkan desain visual menjadi kode *HTML*, *CSS* dan *JavaScript*. Selama program magang ini juga melibatkan proses kolaborasi dengan tim *back-end* untuk integrasi antara UI dengan fungsi-fungsi *server-side*, seperti *API* dan *database*.

Dalam pembentukan *website front-end* pada program magang berikut menggunakan *HTML* dan *CSS*. *HTML* adalah bahasa *markup* yang digunakan untuk membuat struktur dan konten halaman web. Hal ini dapat digunakan untuk menentukan elemen seperti teks, gambar, tautan, tabel, dan formulir pada halaman web[2].

CSS digunakan untuk mengatur tampilan dan gaya elemen *HTML*. Dengan CSS, dapat digunakan untuk mengubah warna, ukuran, margin, dan tata letak elemen pada halaman web[3].

No	Tugas Yang Dilakukan	Minggu	Tanggal Mulai	Tanggal Selesai
1	Pengenalan Perusahaan	1	15 Januari 2024	23 Januari 2024
2	Pembuatan <i>Wireframe Website</i> Perusahaan	2	24 Januari 2024	26 Januari 2024
3	Pembuatan <i>UI/UX Website</i> Perusahaan	3 - 11	29 Januari 2024	19 Maret 2024
4	Pembuatan <i>UI Front-end Website</i> Perusahaan	11 - 13	20 Maret 2024	5 April 2024
5	Pembuatan <i>Back-end Website</i> Perusahaan	15 - 17	15 April 2024	3 Mei 2024
6	Testing	18 - 19	6 Mei 2024	13 Mei 2024
7	<i>Bug Search and Fixing</i>	20	14 Mei 2024	17 Mei 2024

Tabel 3. 1 Timeline Kerja Magang

Aktivitas-aktivitas yang dilakukan selama kerja magang adalah sebagai berikut:

1. Pengenalan Perusahaan (Minggu ke-1)

Dalam minggu pertama magang, para mahasiswa mendapatkan kesempatan unik untuk memperkenalkan diri mereka kepada lingkungan

kerja yang dinamis dan profesional di PT. Nagatama Poin Ciaprima. Selain itu, mereka juga memiliki kesempatan istimewa untuk bertemu langsung dengan beberapa pendiri perusahaan serta anggota tim dari berbagai divisi yang berkontribusi aktif dalam proyek-proyek yang sedang atau akan dilaksanakan. Ini memberikan mereka wawasan yang lebih luas tentang struktur organisasi dan cara kerja perusahaan, serta memberikan pemahaman yang mendalam tentang peran dan tanggung jawab setiap divisi dalam mencapai tujuan bersama.

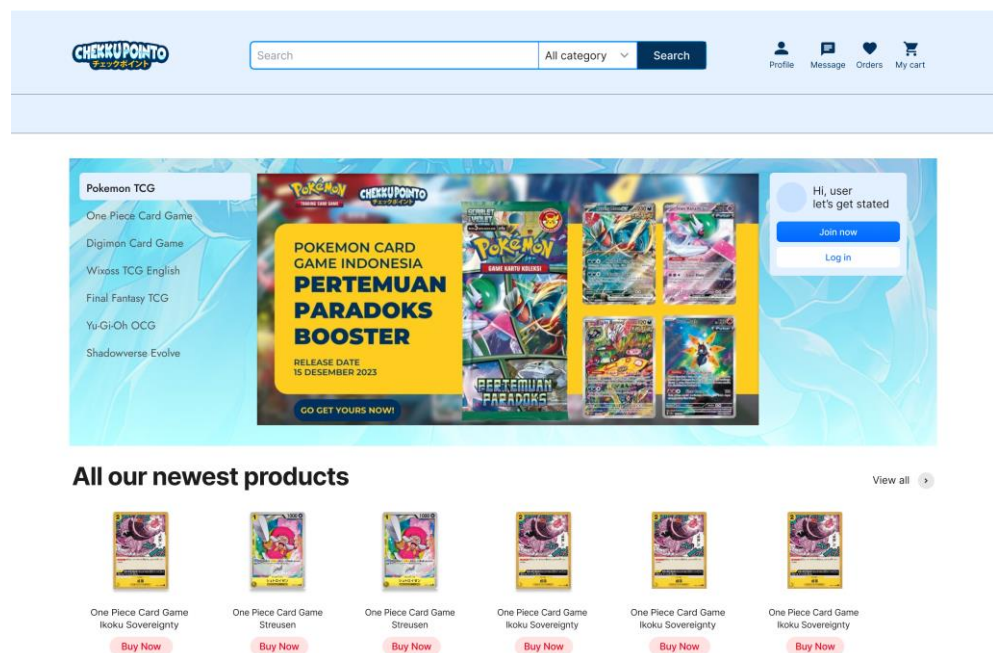
2. Pembuatan *Wireframe Website* Perusahaan (Minggu ke-2)

Pada minggu kedua, proses magang bagi para mahasiswa dilanjutkan dengan fokus pada perancangan *Wireframe* untuk *website* yang sedang dalam pengembangan. Kegiatan ini dilakukan secara kolaboratif antara mahasiswa magang dan beberapa tim atau divisi yang relevan. Tujuan utamanya adalah untuk menyediakan kerangka dasar yang kuat dan jelas untuk *front-end development*, sehingga proses desain dan implementasi menjadi lebih efisien dan efektif. Selain itu, *Wireframe* yang telah dibuat juga berfungsi sebagai alat bantu penting dalam merencanakan desain *website* yang akan dikembangkan di masa mendatang, memungkinkan tim untuk memiliki gambaran yang lebih baik tentang struktur dan tata letak halaman *web* sebelum proses desain visual dimulai.

3. Pembuatan *UI/UX Website* Perusahaan (Minggu ke-3 sampai Minggu ke-11)

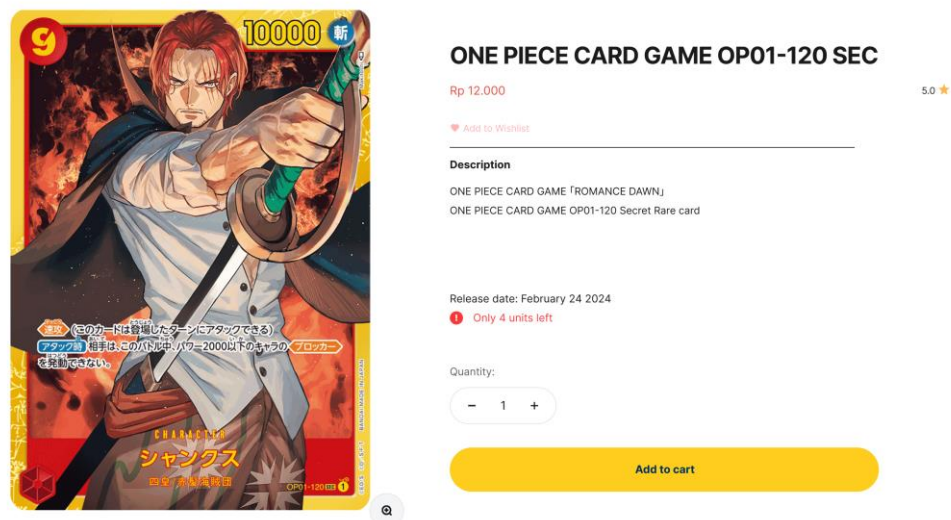
Dalam periode antara minggu ketiga hingga minggu kesebelas, para mahasiswa magang menghabiskan waktu mereka untuk merancang desain *website* perusahaan yang siap untuk digunakan secara luas. Proses ini melibatkan interaksi intensif dengan berbagai divisi internal, termasuk *IT, Network & Infrastructure*, dan *Project Management*, dengan tujuan

utama untuk memastikan bahwa *website* tersebut mudah digunakan bagi pengguna akhir. Diskusi yang berkelanjutan ini penting untuk memastikan bahwa semua elemen *website* dirancang dengan cara yang paling efisien dan efektif, memungkinkan setiap pengguna untuk menavigasi situs dengan mudah.



Gambar 3. 2 Perancangan awal Figma

Pada gambar 3.2, merupakan sebuah *Prototype* yang dibuat sebelum melakukan perancangan terhadap *website* yang akan di buat oleh beberapa divisi yang terkait dalam perancangan *website* ini. *Prototype* ini dibuat dengan menggunakan sebuah *software* bernama *Figma* yang dapat memudahkan berbagai macam divisi untuk melakukan pengecekan maupun pengerjaan secara bersamaan.



Gambar 3. 3 Perancangan Figma

Pada Gambar 3.3, merupakan sebuah tampilan ketika *user* ingin memilih produk yang berada didalam gudang dari PT. Nagatama Poin Ciaprima. Pada tampilan ini ditunjukkan berbagai macam informasi mengenai produk yang dipilih oleh pengguna dari *website* yang dimiliki oleh PT. Nagatama Poin Ciaprima.

Selama fase ini, tim magang menggunakan *software Figma* sebagai alat utama untuk merancang *User Interface (UI)* dan *User Experience (UX)*. Keputusan untuk menggunakan *Figma* didasarkan pada kemampuan aplikasi ini untuk menyimpan proyek secara *real-time* di *cloud*, yang memungkinkan kerja sama tim yang efisien dan kolaboratif. Ini berarti bahwa semua anggota tim dapat berkontribusi pada desain secara simultan dari lokasi yang berbeda, mempercepat proses rancangan dan memastikan bahwa semua perspektif dipertimbangkan sejak awal.

Figma tidak hanya memfasilitasi kerja sama tim yang lebih baik tetapi juga memungkinkan tim untuk menciptakan desain yang responsif dan intuitif, yang penting untuk memberikan pengalaman pengguna yang

positif. Melalui penggunaan *Figma*, tim berhasil menghasilkan desain *website* yang tidak hanya estetik tetapi juga fungsional, memenuhi kebutuhan dan preferensi pengguna akhir.

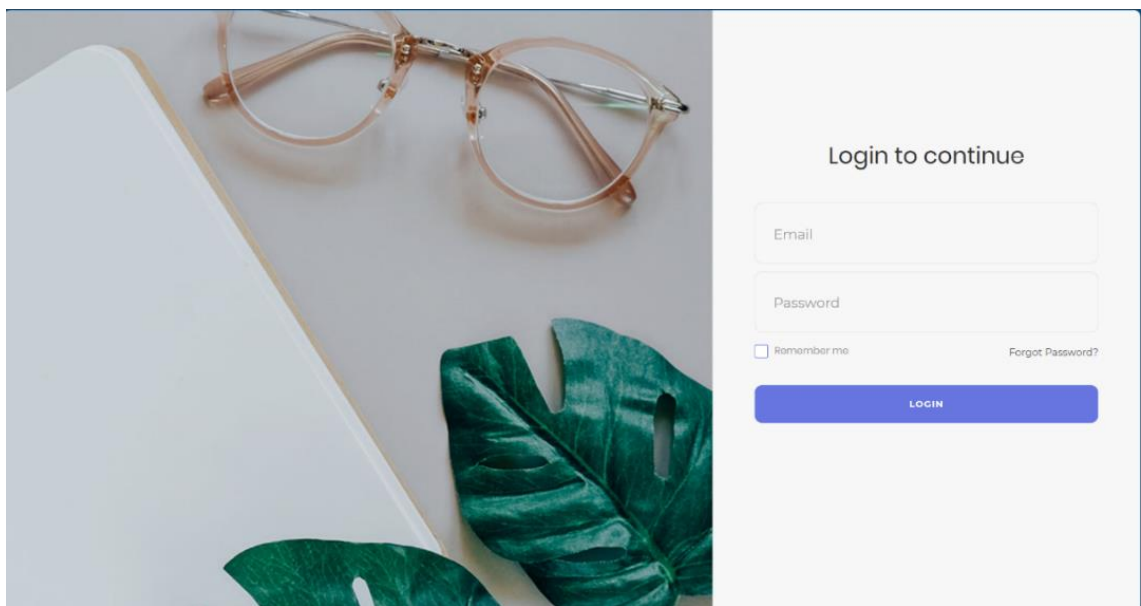
Secara keseluruhan, periode ini menunjukkan betapa pentingnya kerja sama lintas divisi dan penggunaan teknologi modern seperti *Figma* dalam merancang solusi digital yang efektif dan mudah digunakan. Hasilnya adalah desain *website* yang kuat dan fleksibel, siap untuk mendukung operasi perusahaan dan memenuhi kebutuhan pengguna.

4. Pembuatan UI *Front-end Website* Perusahaan

Pada minggu ke-11 hingga ke-13, fokus utama mahasiswa magang adalah pada pembuatan *User Interface (UI) front-end* untuk *website* perusahaan. Proses ini merupakan langkah kritis dalam pengembangan *website*, karena *UI* adalah bagian yang paling dekat dengan pengguna dan memiliki dampak langsung terhadap pengalaman pengguna. Untuk mencapai hasil yang optimal, mahasiswa magang melakukan serangkaian diskusi dan evaluasi yang mendalam tentang desain yang telah direncanakan. Melalui kolaborasi dengan tim *back-end* dan *stakeholder* lainnya untuk memastikan bahwa *UI* yang dibuat sesuai dengan fungsi dan fitur yang diharapkan dari *website*. Selain itu, mahasiswa magang juga mempertimbangkan *feedback* dan masukan dari pengguna potensial untuk memastikan bahwa *website* dapat memenuhi kebutuhan dan harapan pengguna.

Selain itu, sebagai *front-end* mahasiswa magang juga memperhatikan aspek-aspek seperti responsivitas dan aksesibilitas dalam desain *UI* perusahaan. Mahasiswa magang memastikan bahwa *website* dapat diakses dan dinavigasi dengan mudah oleh pengguna dari berbagai perangkat dan kondisi. Hal ini penting untuk memastikan inklusivitas dan ketersediaan *website* kepada audiens yang lebih luas[4]. Pada akhir

minggu ke-13, mahasiswa magang telah berhasil menyelesaikan pembuatan UI *front-end* untuk *website* perusahaan. Mahasiswa magang telah mengimplementasikan desain yang menarik dan fungsional, yang dirancang untuk memberikan pengalaman pengguna yang positif dan memudahkan navigasi pengguna melalui *website*.



Gambar 3. 4 Halaman Login

Pada gambar 3.4 di atas, merupakan halaman *login* yang merupakan penghubung antara *website* yang akan dituju, pada halaman ini, pengguna akan diminta untuk *memasukan* akun yang mereka punya untuk dapat masuk ke dalam *website* yang telah di rancang. Halaman login ini memiliki peran krusial sebagai jembatan bagi setiap pengguna untuk dapat mengakses konten dan fitur yang tersedia dalam suatu platform digital. Gambar 3.4 yang disebutkan sebelumnya menunjukkan visualisasi dari halaman login tersebut, yang merupakan bagian integral dari

arsitektur keseluruhan situs web. Berikut merupakan *code* yang terdapat pada halaman login ini :

```
views > login.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Chekku Pointo Official</title>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <!--
8 <link rel="icon" type="image/png" href="assets/icons/chekku.png"/>
9 <!--
10 <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
11 <!--
12 <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">
13 <!--
14 <link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-font.min.css">
15 <!--
16 <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
17 <!--
18 <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">
19 <!--
20 <link rel="stylesheet" type="text/css" href="vendor/animsition/css/animsition.min.css">
21 <!--
22 <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
23 <!--
24 <link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">
25 <!--
26 <link rel="stylesheet" type="text/css" href="css/util.css">
27 <link rel="stylesheet" type="text/css" href="css/main.css">
28 <!--
29 </head>
30 <body style="background-color: #666666;">
31
32 <div class="limiter">
33 <div class="container-login100">
34 <div class="wrap-login100">
35 <form class="login100-form validate-form" action="process/login.php" method="POST">
36 <span class="login100-form-title p-b-43">
37 Login to continue
38 </span>
39 </div>
40 </div>
41 </div>
```

Gambar 3.5 Code halaman Login

Pada gambar 3.5, merupakan sebuah *code* untuk memasukan file css yang sudah dirancang untuk dapat berfungsi di dalam website

```
41 <div class="wrap-input100 validate-input" data-validate = "Valid email is required: ex@abc.xyz">
42 <input class="input100" type="text" name="email">
43 <span class="focus-input100"></span>
44 <span class="label-input100">Email</span>
45 </div>
46
47
48 <div class="wrap-input100 validate-input" data-validate="Password is required">
49 <input class="input100" type="password" name="pass">
50 <span class="focus-input100"></span>
51 <span class="label-input100">Password</span>
52 </div>
53
54 <div class="flex-sb-m w-full p-t-3 p-b-32">
55 <div class="contact100-form-checkbox">
56 <input class="input-checkbox100" id="ckb1" type="checkbox" name="remember-me">
57 <label class="label-checkbox100" for="ckb1">
58 Remember me
59 </label>
60 </div>
61
62 <div>
63 <a href="#" class="txt1">
64 Forgot Password?
65 </a>
66 </div>
67 </div>
68
69
70 <div class="container-login100-form-btn">
71 <button class="login100-form-btn">
72 Login
73 </button>
74 </div>
75 </form>
76
77 <div class="login100-more" style="background-image: url('assets/img/bg-01.jpg');">
78 </div>
79 </div>
```

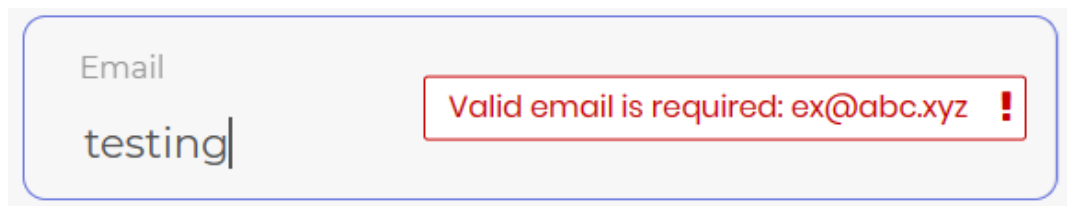
Gambar 3.6 Code halaman Login

Code diatas merupakan *code* untuk mengintegrasikan *CSS*, maupun *javascript* yang terdapat di dalam satu *project* ini, serta merupakan sebuah *code* untuk membuat tampilan *login* seperti di gambar 3.6 , lalu pada *code* ini juga terdapat beberapa *script javascript* yang memiliki tujuan untuk membuat tampilan *website* lebih menarik. Serta pada *code Javascript* juga memberikan sebuah validasi akan *input* yang diberikan oleh pengguna. Jika tidak sesuai dengan *input* yang seharusnya maka akan muncul sebuah *warning* dengan yang sesuai.

```
function validate (input) {  
  if($(input).attr('type') == 'email' || $(input).attr('name') == 'email') {  
    if($(input).val().trim().match(/^([a-zA-Z0-9_\.-]+)@([\d-]{1,3}\.([0-9]{1,3}\.([0-9]{1,3})\.)|([a-zA-Z0-9-]+\.)?([a-zA-Z]{1,5}|[0-9]{1,3})\.$)/) == null) {  
      return false;  
    }  
  }  
  else {  
    if($(input).val().trim() == ''){  
      return false;  
    }  
  }  
}
```

Gambar 3. 7 Code Javascript

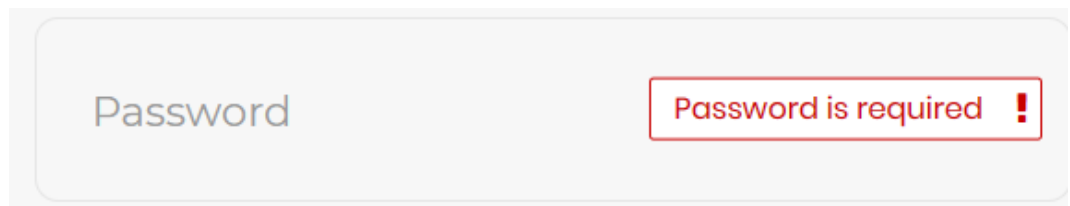
Pada gambar 3.7, merupakan sebuah *code javascript* yang memberikan pemeriksaan terhadap *input* yang diberikan kepada *user*. Pada *script javascript* diatas, berfungsi untuk melakukan validasi terhadap email yang telah di *input* oleh pengguna yang akan dilakukan pengecekan terhadap *input* yang telah diberikan oleh pengguna.



The image shows a web form with an email input field. The input field contains the text "testing|". To the right of the input field, there is a red-bordered box containing the error message "Valid email is required: ex@abc.xyz !" in red text. The entire form is enclosed in a light blue rounded rectangle.

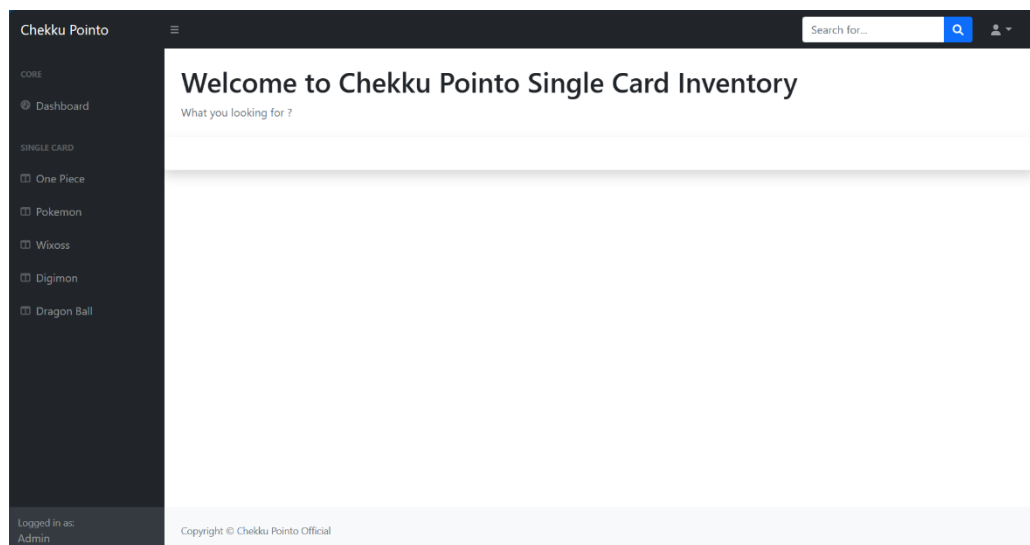
Gambar 3. 8 Peringatan Input Tidak Sesuai

Berikut merupakan sebuah peringatan jika *input* tidak sesuai dan akan diberikan contoh bagaimana semestinya *input* diberikan pada email.



Gambar 3. 9 Tampilan Input

Hal ini juga terjadi pada *input* yang akan berisi sebuah *password* dari pengguna. Jika pengguna tidak memasukan *password* yang pengguna punya maka akan muncul sebuah peringatan bahwa "*Password is required*".



Gambar 3. 10 Halaman Utama

Pada gambar 3.10 diatas , merupakan halaman utama dari *website* yang telah di rancang. Halaman *website* dirancang dengan tema minimalis dan modern dengan tujuan dapat memudahkan beberapa pengguna dalam pemakaian *website* ini. Selain itu, pembuatan *website* ini juga sudah dipertimbangkan untuk digunakan oleh beberapa pengguna agar pengguna dapat dengan mudah menggunakan *website* ini.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>Chekku Pointo Official</title>
    <link rel="icon" type="image/png" href="assets/icons/chekku.png"/>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@latest/dist/style.css" rel="stylesheet" />
    <link href="css/styles.css" rel="stylesheet" />
    <script src="https://use.fontawesome.com/releases/v6.1.0/js/all.js" crossorigin="anonymous"></script>
  </head>
  <body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
      <!-- Navbar Brand -->
      <a class="navbar-brand ps-3" href="..index.php">Chekku Pointo </a>
      <!-- Sidebar Toggle -->
      <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0" id="sidebarToggle" href="#"><i class="fas fa-bars"></i></button>
      <!-- Navbar Search -->
      <form class="d-none d-md-inline-block form-inline ms-auto me-0 me-md-3 my-2 my-md-0">
        <div class="input-group">
          <input class="form-control" type="text" placeholder="Search for..." aria-label="Search for..." aria-describedby="btnNavbarSearch" />
          <button class="btn btn-primary" id="btnNavbarSearch" type="button"><i class="fas fa-search"></i></button>
        </div>
      </form>
      <!-- Navbar -->
      <ul class="navbar-nav ms-auto ms-md-0 me-3 me-lg-4">
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-fw"></i></a>
          <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="index.php?logout"></a></li>
          </ul>
        </li>
      </ul>
    </nav>
  </body>
</html>
```

Gambar 3. 11 Code Bagian Navigation

Pada gambar 3. 11 diatas , merupakan sebuah *code* untuk bagian *navigation bar* pada halaman *website* yang terdapat pada gambar 3.10 . Pada *code* ini juga digunakan beberapa fungsi yang terdapat pada *CSS* yang bertujuan untuk membuat halaman *website* ini lebih menarik dan menjadi lebih interaktif kepada pengguna.

```

38     <div id="layoutSidenav">
39         <div id="layoutSidenav_nav">
40             <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
41                 <div class="sb-sidenav-menu">
42                     <div class="nav">
43                         <div class="sb-sidenav-menu-heading">Core</div>
44                         <a class="nav-link" href="admin.php">
45                             <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
46                             Dashboard
47                         </a>
48                         <div class="sb-sidenav-menu-heading">Single Card</div>
49                         <a class="nav-link" href="admin.php?page=onepiece">
50                             <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
51                             One Piece
52                         </a>
53
54                         <a class="nav-link" href="admin.php?page=pokemon">
55                             <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
56                             Pokemon
57
58                         </a>
59                         <a class="nav-link" href="admin.php?page=wixoss">
60                             <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
61                             Wixoss
62
63                         </a>
64                         <a class="nav-link" href="admin.php?page=digimon">
65                             <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
66                             Digimon
67
68                         </a>
69                         <a class="nav-link" href="admin.php?page=dragonball">
70                             <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
71                             Dragon Ball
72
73                         </a>
74                     </div>
75                 </div>
76                 <div class="sb-sidenav-footer">
77                     <div class="small">Logged in as:</div>
78                     Admin
79                 </div>
80             </nav>
81         </div>
82         <div id="layoutSidenav_content">
83             <main>
84                 <div class="container-fluid px-4">
85                     <h1 class="mt-4">Welcome to Chekku Pointo Single Card Inventory</h1>
86                     <ol class="breadcrumb mb-4">
87                         <li class="breadcrumb-item active">What you looking for ?</li>
88                     </ol>
89                 </div>

```

Gambar 3. 12 Code halaman Utama

Pada gambar 3. 12 diatas, merupakan sebuah *code* untuk membuat menu pada halaman utama *website* yang berada di kiri halaman yang bertujuan untuk masuk ke dalam halaman *website* yang terdapat pada tampilan *website* sebelumnya, pada *code* ini juga menggunakan beberapa *class* yang terdapat pada *website bootstrap* yang bertujuan untuk memudahkan mahasiswa magang dalam mengerjakan *project* ini.

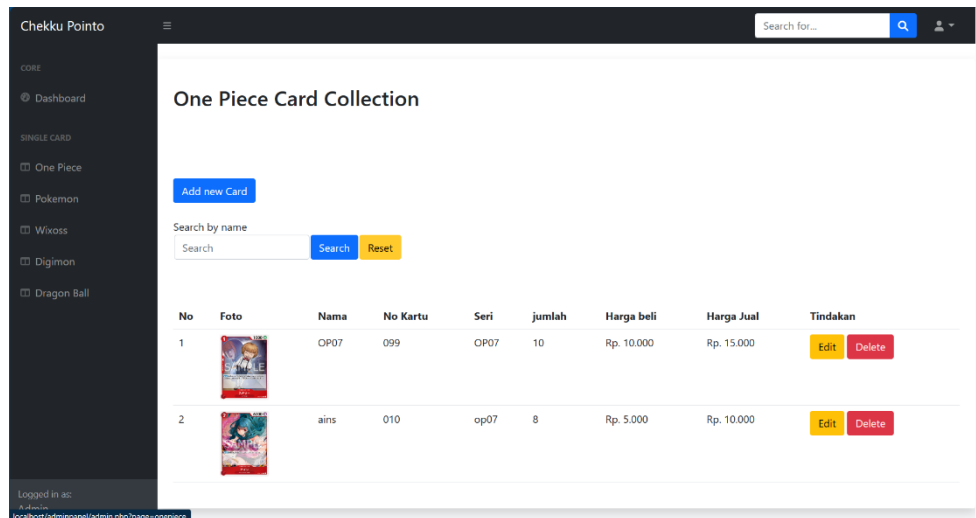
```

<div id="layoutSidenav_content">
  <main>
    <div class="container-fluid px-4">
      <h1 class="mt-4">Welcome to Chekku Pointo Single Card Inventory</h1>
      <ol class="breadcrumb mb-4">
        <li class="breadcrumb-item active">What you looking for ?</li>
      </ol>
    </div>
    <main role="main" class="card border-0 rounded shadow p-4">
      <?php
      if(!isset($_GET['page'])){
        $page = "index";
      }else {
        $page = $_GET['page'];
      }
      require 'viewss/'. $page . '.php';
      ?>
    </main>
  </main>
  <footer class="py-4 bg-light mt-auto">
    <div class="container-fluid px-4">
      <div class="d-flex align-items-center justify-content-between small">
        <div class="text-muted">Copyright &copy; Chekku Pointo Official</div>
      </div>
    </div>
  </footer>
</div>

```

Gambar 3. 13 Code untuk mengganti halaman

Pada gambar 3.13 diatas, merupakan sebuah *code* untuk mengganti halaman sesuai dengan menu yang dipilih oleh pengguna. Pada *code* ini, mahasiswa magang menggunakan sebuah fungsi *PHP* untuk mengubah beberapa halaman dengan menggunakan fungsi *if else* di dalamnya. Hal ini juga dapat membuat *website* lebih efisien untuk digunakan serta dengan menerapkan fungsi *PHP* ini, dapat memudahkan pengembang *website* untuk dapat menambah halaman *website* didalamnya.



Gambar 3.14 Tampilan Halaman One Piece

Pada gambar 3.14 diatas merupakan sebuah tampilan ketika pengguna memilih halaman *one Piece* pada halaman ini akan menunjukkan segala *product* yang mempunyai hubungan dengan *one piece*. Pada halaman ini juga pengguna dapat menambahkan sebuah *product* baru didalamnya, selain menambahkan *product* baru, pengguna juga dapat melakukan perubahan terhadap *product* yang telah dimasukan kedalamnya. Jika terjadi sebuah kesalahan, pengguna juga dapat menghapus item yang telah dimasukan kedalamnya. Pengguna juga dapat melakukan pencaharian terhadap *product* yang ingin dicari dengan mencarinya di dalam *search box* yang telah di berikan di bagian atas. Jika pengguna telah melakukan pencaharian dan ingin kembali ke halaman awal, pengguna dapat menggunakan tombol reset yang berada di samping tombol pencaharian.


```

<?php
require 'function.php';
?>

<h2 class = "mt-3 h2">One Piece Card Collection</h2>
<br>
<br>

<br>
<br>

<div class="form-group has-search">
  <table>
    <tr>
      <a href = "admin.php?page=onepiece-form&action=add" class = "btn btn-primary">
      <span data-feather="plus-circle"></span> Add new Card</a>
      <br>

      <br>
      <tr>Search by name</tr>
      <td>
        <form action="admin.php?page=onepiece" method="get">
          <input type="hidden" name="page" value="onepiece">
          <input type="text" class="form-control" name="search" placeholder="Search">
        </td>
        <td>
          <button type="submit" class="btn btn-primary">Search</button>
        </form>
        </td>
        <td>
          <a href="admin.php?page=onepiece" class="btn btn-warning">Reset</a>
        </td>
      </tr>
    </table>
  </div>
<br>
<br>

<div class = "table-responsive mt-3">
  <table class = "table">
    <tr>
      <th>No</th>
      <th>Foto</th>
      <th>Nama</th>
      <th>No Kartu</th>
      <th>Seri</th>
      <th>jumlah</th>
      <th>Harga beli</th>
      <th>Harga Jual</th>
      <th>Tindakan</th>
    </tr>

```

Gambar 3. 15 Code Halaman Website

Pada gambar 3.15, merupakan sebuah *code* untuk halaman *website* yang terdapat pada gambar 3.15 , pada *code* ini, mahasiswa magang menggunakan sebuah fungsi *table* yang bertujuan untuk memudahkan pengguna dalam melihat *product* yang terdapat pada halaman *website* ini.

The image shows a web application interface for adding a One Piece collection item. The page title is "Tambah One Piece Collection". The form includes the following fields:

- Link Foto
- Name
- No Kartu
- Seri
- Jumlah
- Harga Beli (Rp)
- Harga Jual (Rp)

A green "Simpan" button is located at the bottom of the form. The sidebar on the left shows navigation options: Dashboard, One Piece, Pokemon, Wixoss, Digimon, and Dragon Ball. The user is logged in as Admin.

Gambar 3. 16 Tampilan halaman formulir

Pada gambar 3.16 diatas, merupakan sebuah tampilan ketika pengguna *website* ini ingin menambahkan sebuah *product* yang berkaitan dengan *product one piece*. Pada halaman ini sudah disediakan beberapa formulir yang diperlukan untuk menambah *product* tersebut. Mahasiswa magang juga membuat tampilan ini agar mudah dimengerti oleh semua pengguna. Format foto yang diberikan juga berbentuk sebuah *link* dikarenakan dari perusahaan sudah terbiasa untuk menyimpan foto di *website*, maka dari itu akan memudahkan pengembang dalam melakukan *upload file* kedalam *database* dan memunculkannya kedalam halaman *website*

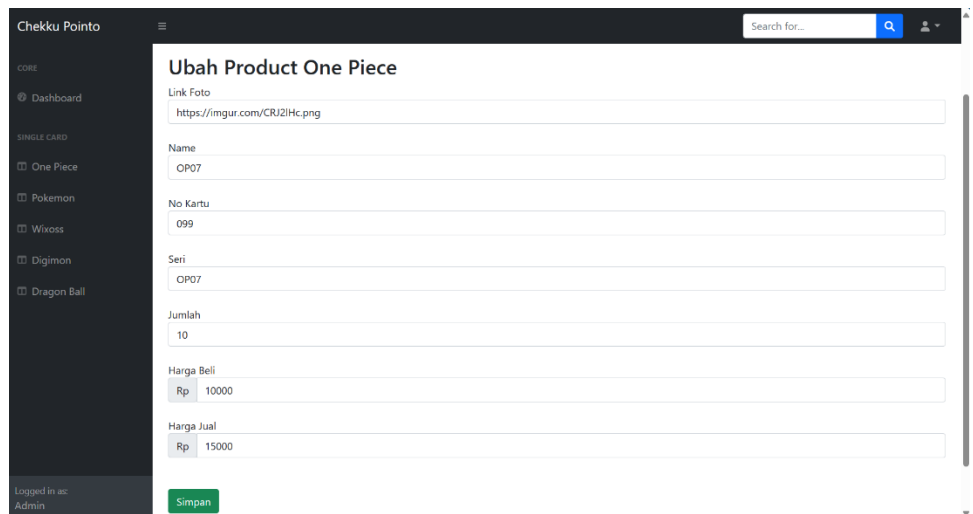
```

43
44 <h1 class = "h2 mt-3">= $page_title; ?&gt;&lt;/h1&gt;
45 &lt;form action = "process/onepiece.php?action=&lt;?=$action; ?&gt;" method = "post"&gt;
46     &lt;div class = "form-group"&gt;
47         &lt;label&gt; Link Foto&lt;/label&gt;
48         &lt;input type = "text" name = "link" class = "form-control" value = "&lt;?=$link; ?&gt;"&gt;
49     &lt;/div&gt;
50     &lt;br&gt;
51     &lt;div class = "form-group"&gt;
52         &lt;label&gt; Name&lt;/label&gt;
53         &lt;input type = "text" name = "name" class = "form-control" value = "&lt;?=$name; ?&gt;"&gt;
54     &lt;/div&gt;
55     &lt;br&gt;
56     &lt;div class = "form-group"&gt;
57         &lt;label&gt; No Kartu&lt;/label&gt;
58         &lt;input type = "text" name = "code" class = "form-control" value = "&lt;?=$noKartu; ?&gt;"&gt;
59     &lt;/div&gt;
60     &lt;br&gt;
61     &lt;div class = "form-group"&gt;
62         &lt;label&gt; Seri&lt;/label&gt;
63         &lt;input type = "text" name = "series" class = "form-control" value = "&lt;?=$series; ?&gt;"&gt;
64     &lt;/div&gt;
65     &lt;br&gt;
66     &lt;div class = "form-group"&gt;
67         &lt;label&gt; Jumlah&lt;/label&gt;
68         &lt;input type = "text" name = "jumlah" class = "form-control" value = "&lt;?=$jumlah; ?&gt;"&gt;
69     &lt;/div&gt;
70     &lt;br&gt;
71     &lt;div class="form-group"&gt;
72         &lt;label&gt;Harga Beli&lt;/label&gt;
73         &lt;div class="input-group"&gt;
74             &lt;div class="input-group-prepend"&gt;
75                 &lt;span class="input-group-text"&gt;Rp&lt;/span&gt;
76             &lt;/div&gt;
77             &lt;input type="text" name="hargaBeli" class="form-control" value="&lt;?=$hargaBeli; ?&gt;"&gt;
78         &lt;/div&gt;
79     &lt;/div&gt;
80     &lt;br&gt;
81     &lt;div class="form-group"&gt;
82         &lt;label&gt;Harga Jual&lt;/label&gt;
83         &lt;div class="input-group"&gt;
84             &lt;div class="input-group-prepend"&gt;
85                 &lt;span class="input-group-text"&gt;Rp&lt;/span&gt;
86             &lt;/div&gt;
87             &lt;input type="text" name="hargaJual" class="form-control" value="&lt;?=$hargaJual; ?&gt;"&gt;
88         &lt;/div&gt;
89 &lt;/div&gt;
90 &lt;br&gt;
91 &lt;br&gt;
92 &lt;input type = "hidden" name = "id" value = "&lt;?=$id; ?&gt;"/&gt;
93 &lt;input type = "hidden" name = "catagory" value = "&lt;?=$catagory; ?&gt;"/&gt;
94 &lt;button type = "submit" class = "btn btn-success"&gt;Simpan&lt;/button&gt;
95 &lt;/form&gt;
</pre

```

Gambar 3. 17 Code Product

Pada gambar 3.17 diatas, merupakan sebuah *code* untuk menampilkan halaman yang digunakan untuk menambahkan sebuah product didalamnya. *Code* yang berada pada gambar 3.17 digunakan juga pada saat pengguna ingin melakukan perubahan kepada *product* yang membutuhkan perubahan. Dengan melakukan hal ini juga dapat memudahkan pengembang *website* agar bekerja lebih efektif dan menghemat waktu karena satu halaman *website* dapat digunakan lebih dari 1 fungsi dan dapat digunakan di beberapa halaman yang di butuhkan



Gambar 3. 18 Tampilan mengubah product

Pada gambar 3.18 diatas , merupakan tampilan untuk melakukan sebuah perubahan di dalam *product* yang akan diubah, dalam halaman ini akan menunjukkan product yang ingin dirubah. Halaman *website* ini, mengambil value dari *product* yang akan dilakukan perubahan terhadap nilai didalamnya. Pengambilan nilai dari *database* untuk ditampilkan kepada halaman *website* adalah dengan menggunakan fungsi *select* pada *PHP* yang akan menarik sebuah nilai dari *database*, lalu akan ditampilkan kedalam halaman *website*. Perbedaan yang terdapat hanya berada pada susunan *code* yang berada saat menambahkan *product* dan berada pada saat ingin mengubah produk yang sudah ada.

5. Pembuatan *Back-end Website* Perusahaan

```
1  <?php
2
3  function connection(){
4      try{
5          $dsn = "mysql:host=localhost;dbname=chekkutest";
6          $user = "root";
7          $pass = "";
8          $pdo = new PDO($dsn, $user, $pass);
9          return $pdo;
10
11     }catch(PDOException $e){
12         return $e;
13     }
14 }
15
16
```

Gambar 3. 19 Code untuk database

Pada gambar 3.19 diatas, merupakan sebuah *code* untuk menghubungkan *website* dengan *database*, untuk *project* yang sedang dikerjakan menggunakan *database dummy* yang memiliki tujuan untuk tidak mengubah *database* yang sudah ada. Pada *code* ini, mahasiswa magang menggunakan sebuah fungsi yang sudah ada di *PHP* seperti *PDO()*. Fungsi *PDO()* merupakan sebuah fungsi yang terdapat dalam bahasa pemograman *PHP* yang memiliki sebuah fungsi untuk menyediakan sebuah antarmuka yang bersifat konsisten yang dapat memudahkan mahasiswa magang untuk berinteraksi dengan berbagai jenis *database*. Dalam hal ini, menggunakan fungsi *PDO* juga dapat membantu dalam menciptakan sebuah *code* yang lebih portabel dan lebuah aman karena dapat membantu mahasiswa magang untuk berpindah dari satu *database* ke *database* lainnya dengan sangat mudah dan tanpa perlu mengubah banyak *code* didalamnya

```

1 <?php
2 require '../function.php';
3 $pdo = connection();
4
5
6 if ($_SERVER["REQUEST_METHOD"] == "POST" || $_SERVER["REQUEST_METHOD"] == "GET") {
7
8     if(isset($_POST['email']) && isset($_POST['pass'])) {
9         $email = $_POST['email'];
10        $password = $_POST['pass'];
11
12    } elseif(isset($_GET['email']) && isset($_GET['pass'])) {
13
14        $email = $_GET['email'];
15        $password = $_GET['pass'];
16
17    } else {
18
19        echo "Email or password not provided";
20        exit();
21    }
22
23
24    $sql = "SELECT * FROM user WHERE email = ? AND password = ?";
25    $stmt = $pdo->prepare($sql);
26
27
28    $stmt->bindValue(1, $email);
29    $stmt->bindValue(2, $password);
30
31    $stmt->execute();
32
33    $result = $stmt->fetch();
34
35    if ($result) {
36
37        header("Location: ../admin.php");
38        exit();
39    } else {
40
41        header("Location: viewss/login.php");
42        exit();
43    }
44
45    $stmt->closeCursor();
46 } else {
47
48    echo "Method not allowed";
49 }
50
51

```

Gambar 3. 20 Code login ke halaman website

Pada gambar 3.20 diatas terdapat *code* untuk melakukan sebuah proses *login* yang menghubungkan antara pengguna dan halaman *website*. Dalam proses *login* ini, mahasiswa magang menggunakan bahasa pemograman *PHP* untuk melakukan proses *login* ini. Mahasiswa magang menggunakan sebuah fungsi *if else* untuk melakukan pengecekan terhadap “Request Method” untuk mengetahui respon dari *website* itu bersifat *post* atau *get*. Setelah itu dilakukan sebuah pengecekan terhadap nilai yang diinput

dengan menggunakan fungsi *isset()*. Dalam fungsi *isset()* bertugas untuk melakukan pengecekan atas nilai yang berada pada *variable* yang ditentukan dengan hal ini akan mencegah sebuah *value* bersifat *null* dalam suatu *database*. Setelah itu akan di lakukan sebuah proses pengecekan melalui kode *SQL* yang bertugas untuk melakukan pengecekan terhadap data yang berada dalam *database*. Jika nilai yang dicari di dalam *database* sesuai dengan apa yang diinput oleh pengguna. Jika nilai yang sesuai dengan yang tertera di dalam *database* maka pengguna akan langsung diarahkan untuk ke *website* utama untuk dapat mengakses *website* dari perusahaan.

```
<?php
    if(!isset($_GET['page'])){
        $page = "index";
    }else {
        $page = $_GET['page'];
    }
    require 'viewss/'. $page . '.php';
?>
</main>
```

Gambar 3. 21 Code page halaman website

Pada gambar 3.21, terdapat sebuah *code* yang digunakan untuk dapat membuka *page* yang diinginkan sesuai dengan kebutuhan pengguna, dengan menggunakan *code* ini dapat memudahkan pengembang dalam melakukan penambahan halaman *website* karena tidak harus mengubah tampilan awal dari *website* yang telah ada. Lalu dengan adanya *code* ini juga akan membuat *website* lebih optimal dan efisien saat digunakan. Pada *code* ini juga menggunakan sebuah fungsi *isset* yang berfungsi untuk melakukan pengecekan terhadap nilai yang di minta.

```

if(isset($_GET['search'])){
    if($_GET['search'] != ""){
        $search = $_GET['search'];
        $pdo = connection();
        $sql = "SELECT p.*, c.CatagoryID
        FROM product p
        JOIN catagory c ON p.CatagoryID = c.CatagoryID
        WHERE p.CatagoryID = 1
        AND (p.Name LIKE :search
        OR p.Code LIKE :search
        OR p.Series LIKE :search)";

        $result = $pdo->prepare($sql);
        $result->execute(['search' => '%'.$search.'%']);
        $data = $result->fetchAll();

        $no = 1;
        foreach((array)$data as $product){?>
            <tr>
                <td><?=$no++ ?></td>
                <td><img src= "<?=$product['Image']?>" alt = "Image" width= "75" height = "100"></td>
                <td><?=$product['Name'] ?></td>
                <td><?=$product['Code'] ?></td>
                <td><?=$product['Series'] ?></td>
                <td><?=$product['Quantity'] ?></td>
                <td>Rp. <?=$number_format($product['HargaBeli'], 0, ',', '.') ?></td>
                <td>Rp. <?=$number_format($product['HargaJual'], 0, ',', '.') ?></td>
                <td>
                    <a href = "admin.php?page=onepiece-form&action=edit&id=<?=$product['ProductID'] ?>" class = "btn btn-warning">
                    <span data-feather="edit"></span> Edit</a>
                    <a href = "process/onepiece.php?action=delete&id=<?=$product['ProductID'] ?>" class = "btn btn-danger"
                    onclick = "return confirm('Apakah anda yakin ingin menghapus data ini?')">
                    <span data-feather="trash-2"></span> Delete</a>
                </td>
            </tr>
        </php
    }
}

```

Gambar 3. 22. Code pencaharian.

Pada gambar 3.22, terdapat sebuah *code* untuk melakukan pencaharian terhadap *product* yang berada di dalam *database*. Pada *code* ini, mahasiswa magang menggunakan fungsi *if else* untuk melakukan pengecekan apakah pengguna ingin melakukan pencarian dengan menggunakan fungsi *isset()* yang bertujuan untuk mengetahui apakah dalam *variable search* ada nilai di dalamnya atau tidak. Jika setelah dilakukan pengecekan terhadap *value* yang ada maka nilai dari *search* akan di tampung *kedalam* sebuah *variable \$search*. Setelah itu akan dilakukan sebuah pencarian dengan menggunakan *query SQL select* untuk mencari *product* yang telah di masukan oleh pengguna. Pada *query SQL* dilakukan sebuah *join* dikarenakan dalam *database* tabel *product* dan *catagory* merupakan 2 tabel yang berbeda.

Setelah dilakukan *join* terhadap tabel *catagory* ke *product*, mahasiswa magang menggunakan sebuah *query where* yang bertujuan untuk mencari nilai yang sedang dicari pengguna. Setelah itu dilakukan maka nilai dari *variable search* akan dicari di dalam *database*. Jika nilai

antara *variabel* dan nilai yang berada dalam *database* sesuai maka akan ditampilkan dengan menggunakan *foreach*. Fungsi *foreach* dalam *PHP* memiliki tugas untuk melakukan sebuah pengulangan untuk setiap elemen yang berada dalam suatu objek. Setelah itu nilai yang telah dilakukan pengulangan oleh fungsi *foreach* akan ditampilkan kedalam *website* dengan menggunakan format *table*. Mahasiswa magang juga menggunakan *number_format* untuk membuat angka pada harga beli dan harga jual memiliki titik.

```

$pdo = connection();
$sql = "SELECT p.*, c.CatagoryID FROM product p
      JOIN catagory c ON p.CatagoryID = p.CatagoryID
      Where p.CatagoryID = 1"
      ;
$result = $pdo->prepare($sql);
$result->execute();
$data = $result->fetchAll();

$no = 1;
foreach((array)$data as $product){?>
<tr>
<td><?=$no++ ?></td>
<td></td>
<td><?=$product['Name'] ?></td>
<td><?=$product['Code'] ?></td>
<td><?=$product['Series'] ?></td>
<td><?=$product['Quantity'] ?></td>
<td>Rp. <?=$number_format($product['HargaBeli'], 0, ',', '.') ?></td>
<td>Rp. <?=$number_format($product['HargaJual'], 0, ',', '.') ?></td>
<td>
<a href="admin.php?page=onepiece-form&action=edit&id=<?=$product['ProductID'] ?>" class="btn btn-warning">
<span data-feather="edit"></span> Edit</a>

<a href="process/onepiece.php?action=delete&id=<?=$product['ProductID'] ?>" class="btn btn-danger"
onclick="return confirm('Apakah anda yakin ingin menghapus data ini?')">
<span data-feather="trash-2"></span> Delete</a>
</td>

<?php
}

```

Gambar 3. 23 Jika tidak melakukan pencaharian

Pada gambar 3.23, merupakan sebuah *code* jika pengguna tidak ingin melakukan sebuah pencaharian. Dengan *code* ini akan menampilkan segala data yang berhubungan dengan *one piece* didalamnya. Pada hal ini juga, mahasiswa magang menggunakan sebuah *query select* pada *SQL* dengan tujuan untuk dapat mengambil semua nilai yang berhubungan dengan halaman yang telah digunakan oleh pengguna. Setelah itu mahasiswa magang juga melakukan injeksi terhadap beberapa *code* html

dengan cara memasukan *code PHP* juga di dalamnya agar semua nilai yang berada dalam *database* bisa ditampilkan secara menyeluruh.

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

$action = $_GET['action'];

if($_GET['action'] == 'add'){
    $page_title = 'Tambah One Piece Collection';
    $link = "";
    $name = "";
    $catagory = "1";
    $noKartu = "";
    $code = "";
    $series = "";
    $jumlah = "";
    $hargaBeli = "";
    $hargaJual = "";
    $id = "";
}
```

Gambar 3. 24. Code PHP

Pada gambar 3.24, merupakan sebuah *code PHP* untuk halaman formulir yang digunakan dalam menambah maupun melakukan perubahan terhadap nilai yang berada didalam *database*. *Code PHP* yang terdapat pada gambar 3.24, dibiarkan tetap kosong agar formulir yang digunakan untuk menambahkan dapat di isi dengan nilai yang baru. Penggunaan variable *page_title* bertujuan untuk mengubah judul pada formulir yang akan digunakan. Mahasiswa magang juga menggunakan fungsi *if else* untuk mengetahui *action* yang telah di *request* oleh pengguna.

```

18
19 }else if($_GET['action'] == 'edit'){
20
21     $page_title = 'Ubah Product One Piece';
22
23     $pdo = connection();
24
25     $sql = "SELECT * FROM product WHERE ProductID = ?";
26     $hasil = $pdo -> prepare($sql);
27     $hasil -> execute($_GET['id']);
28     $row = $hasil -> fetch();
29     $link = $row['Image'];
30     $name = $row['Name'];
31     $category = "1";
32     $noKartu = $row['Code'];
33     $series = $row['Series'];
34     $jumlah = $row['Quantity'];
35     $hargaBeli = $row['HargaBeli'];
36     $hargaJual = $row['HargaJual'];
37     $id = $row['ProductID'];
38
39
40 }
41 ?>

```

Gambar 3. 25. Code pengisian nilai

Pada gambar 3. 25 diatas , terdapat *code* yang memiliki fungsi untuk melakukan pengisian nilai kepada formulir. Mahasiswa magang melakukan sebuah fungsi untuk dapat mengambil nilai dari *database* yang sesuai dengan yang ingin diubah oleh pengguna. Segala *value* yang sudah di dapatkan dari *database* akan di pindahkan kedalam formulir yang dapat dilihat oleh pengguna.

```

<h1 class = "h2 mt-3"><?= $page_title; ?></h1>
<form action = "process/onepiece.php?action=<?= $action; ?>" method = "post">
  <div class = "form-group">
    <label> Link Foto</label>
    <input type = "text" name = "link" class = "form-control" value = "<?= $link; ?>">
  </div>
  <br>
  <div class = "form-group">
    <label> Name</label>
    <input type = "text" name = "name" class = "form-control" value = "<?= $name; ?>">
  </div>
  <br>
  <div class = "form-group">
    <label> No Kartu</label>
    <input type = "text" name = "code" class = "form-control" value = "<?= $noKartu; ?>">
  </div>
  <br>
  <div class = "form-group">
    <label> Seri</label>
    <input type = "text" name = "series" class = "form-control" value = "<?= $series; ?>">
  </div>
  <br>
  <div class = "form-group">
    <label> Jumlah</label>
    <input type = "text" name = "jumlah" class = "form-control" value = "<?= $jumlah; ?>">
  </div>
  <br>
  <div class="form-group">
    <label>Harga Beli</label>
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text">Rp</span>
      </div>
      <input type="text" name="hargaBeli" class="form-control" value="<?= $hargaBeli; ?>">
    </div>
  </div>
  <br>
  <div class="form-group">
    <label>Harga Jual</label>
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text">Rp</span>
      </div>
      <input type="text" name="hargaJual" class="form-control" value="<?=$hargaJual; ?>">
    </div>
  </div>
</div>

```

Gambar 3. 26. Code PHP untuk mengisi nilai sesuai dengan permintaan pengguna

Pada gambar 3.26 diatas merupakan *code front-end* yang telah diintegrasikan dengan *PHP* yang berfungsi untuk mengisi nilai” sesuai dengan permintaan pengguna. Jika pengguna ingin menambahkan sebuah produk maka akan terkoneksi dengan *back-end* yang menunjukkan *action add* sama dengan jika pengguna ingin melakukan perubahan tampilan formulir akan sesuai dengan permintaan pengguna.

```

require '../function.php';
$pdo = connection();

if ($_GET['action'] == 'add') {
    $sql = "INSERT INTO product (CatagoryID, Image, Name, Code, Series, Quantity, HargaBeli, HargaJual)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = $pdo -> prepare($sql);
    $stmt -> execute([
        $_POST['catagory'],
        $_POST['link'],
        $_POST['name'],
        $_POST['code'],
        $_POST['series'],
        $_POST['jumlah'],
        $_POST['hargaBeli'],
        $_POST['hargaJual']
    ]);

    header('Location: (admin.php?page=product)');
}

```

Gambar 3. 27 Code menambah product

Pada gambar 3.27 diatas merupakan *code* yang digunakan untuk menambah *product* ke dalam *database*. Maka dari itu, mahasiswa magang membuat sebuah *query insert* pada *SQL* yang bertujuan untuk dapat memasukkan semua nilai yang ingin dimasukkan ke dalam *database*. Namun sebelum memasukkan nilai ke dalam *database* akan dilakukan sebuah pengecekan terhadap aksi yang dilakukan oleh *user*. Setelah dilakukan pengecekan maka akan dilakukan pengambilan nilai dengan menggunakan fungsi *PHP* yaitu *\$_POST* agar dapat mengambil nilai dari halaman *website* sebelumnya. Setelah mengambil nilai dari formulir yang berada pada halaman *website* sebelumnya, akan dilakukan sebuah *pros .es* untuk memasukkan nilai yang sudah didapatkan dari halaman *website* untuk dapat dimasukkan ke dalam *database* yang sudah ada ke dalam *database* yang sudah tersedia. Sebelum melakukan hal itu, mahasiswa magang telah melakukan beberapa observasi untuk membuat perancangan *back-end* ini sesuai rencana dan dapat membuat *website* yang telah dirancang dapat berjalan dengan baik dan tidak terjadi sebuah *error* yang merugikan bagi perusahaan maupun bagi pengguna yang akan menggunakan *website* ini. Setelah proses menambahkan sudah terjadi dan sudah berhasil maka setelah *system* akan langsung mengarahkan pengguna

kepada halaman yang dia pilih sebelumnya. Dengan hal ini juga dapat membuat pengguna lebih nyaman dalam menggunakan *website* dari perusahaan ini.

```
} else if ($_GET['action'] == 'edit') {
    $sql = "UPDATE product
           SET Image= ?,
             Name = ?,
             Code = ?,
             Series = ?,
             Quantity = ?,
             HargaBeli = ?,
             HargaJual = ?
           WHERE ProductID = ?";

    $stmt = $pdo -> prepare($sql);
    $stmt ->execute ([
        $_POST['link'],
        $_POST['name'],
        $_POST['code'],
        $_POST['series'],
        $_POST['jumlah'],
        $_POST['hargaBeli'],
        $_POST['hargaJual'],
        $_POST['id']
    ]);
    header('Location: ../admin.php?page=onepiece');
```

Gambar 3. 28 Code perubahan terhadap product

Pada gambar 3.28, merupakan sebuah *code* yang digunakan pada halaman *website* untuk melakukan sebuah perubahan terhadap *product* yang telah berada di dalam *database*. Maka dari itu, mahasiswa magang membuat sebuah *query update* pada *SQL* yang bertujuan untuk mengubah

nilai yang berada pada *database*. Namun sebelum melakukan sebuah perubahan akan suatu *product* di dalam *database* akan dilakukan sebuah pengecekan terhadap nilai yang ingin dirubah dengan menggunakan fungsi *\$_POST* dalam *PHP* yang bertujuan untuk mengambil sebuah nilai yang terdapat pada halaman *website* yang berisi dengan formulir yang ada pada page sebelumnya. Setelah itu, akan dilakukan sebuah pengambilan nilai yang berasal dari halaman *website* sebelumnya, akan dilakukan sebuah proses untuk memasukan nilai yang sudah didapatkan dari halaman *website* untuk dapat merubah nilai yang berada pada *database*. Sebelum melakukan hal itu, mahasiswa magang telah melakukan observasi dalam hal pembuatan *back-end* ini. Para mahasiswa magang telah mempertimbangkan beberapa masalah yang telah diprediksi dalam pengerjaan *project website* perusahaan ini. Maka dari itu, para mahasiswa magang sudah memberikan sebuah metode yang paling efektif serta lebih efisien untuk pengerjaan *website* ini. Ketika proses perubahan data telah dilakukan maka pengguna akan diarahkan untuk menuju tampilan *website* yang sebelumnya telah digunakan oleh pengguna *website*

```

}else if ($_GET['action'] == 'delete') {

    $sql = "DELETE FROM product
           WHERE ProductID = ?";

    $stmt = $pdo->prepare($sql);
    $stmt->execute([$_GET['id']]);

    header('Location: ../admin.php?page=onepiece');
}

```

Gambar 3.29 Code penghapusan product

Pada gambar 3.29, merupakan sebuah *code* untuk melakukan penghapusan terhadap *product* yang berada pada *database*. Maka dari itu, mahasiswa magang membuat sebuah *query delete* pada *SQL* yang memiliki tujuan untuk dapat menghapus sebuah nilai yang berada pada *database*. Namun sebelum melakukan aksi untuk menghapus data yang berada di dalam *database*, mahasiswa magang membuat *code* untuk mengecek aksi yang di berikan oleh user kedalam *website*. Setelah dilakukan pengecekan terhadap aksi yang telah diterima oleh pengguna, maka akan dilakukan sebuah fungsi *\$_GET* untuk mengambil nilai dari *product* yang ingin di hapus. Dengan hal ini, membuat *website* yang dijalankan menjadi lebih efisien dan hemat daya. Ketika pengguna ingin melakukan sebuah aksi untuk melakukan penghapusan terhadap nilai yang berada didalam *database* maka akan muncul notifikasi untuk melakukan konfirmasi kepada pengguna, ketika pengguna telah melakukan konfirmasi maka akan dijalankan sebuah proses untuk melakukan penghapusan terhadap nilai yang berada di dalam *database*. Setelah dilakukan

penghapusan nilai di *database*, pengguna akan diarahkan untuk pindah ke halaman *website* yang sebelumnya telah di akses oleh pengguna.

6. *Testing*

Pada minggu ke-18 hingga ke-19, mahasiswa magang mengarahkan upaya mereka ke tahap *testing*, yang merupakan langkah kritis dalam siklus pengembangan perangkat lunak. Tahap ini dirancang untuk memvalidasi fungsi dan fitur *website* yang telah dirancang dan dikembangkan selama periode sebelumnya. Tujuannya adalah untuk memastikan bahwa semua komponen *website* berfungsi sesuai dengan spesifikasi dan standar kualitas yang ditetapkan.

Tahapan *testing* ini melibatkan serangkaian aktivitas yang dirancang untuk menemukan dan memperbaiki bug atau masalah lainnya sebelum *website* diluncurkan. Ini termasuk pengujian unit, integrasi, sistem, dan pengguna, serta pengujian keamanan dan performa. Tujuannya adalah untuk memastikan bahwa *website* dapat menangani beban penggunaan yang diharapkan dan memberikan pengalaman pengguna yang optimal[5].

Selama tahapan ini, tim pengembangan menggunakan berbagai alat dan teknik pengujian, termasuk *automated testing tools*, *manual testing*, dan metode pengujian berbasis skenario. Tim pengembangan juga mungkin melakukan pengujian *cross-browser* untuk memastikan kompatibilitas *website* dengan berbagai jenis *browser* dan *platform*. Secara keseluruhan, tahapan *testing* pada minggu ke-18 hingga ke-19 adalah kunci untuk memastikan bahwa *website* yang sedang dikembangkan memenuhi standar kualitas tinggi dan siap untuk digunakan oleh pengguna akhir[6].

7. *Bug Search and Fixing*

Pada minggu ke-20, tim magang fokus pada aktivitas *Bug Search and Fixing*, yang merupakan tahapan akhir dalam siklus pengembangan perangkat lunak. Tahapan ini melibatkan pencarian, identifikasi, dan pemecahan *bug* atau masalah yang mungkin ada dalam kode sumber atau aplikasi yang sedang dikembangkan. Tujuannya adalah untuk meningkatkan kualitas produk dan memastikan bahwa semua fitur berfungsi sesuai dengan spesifikasi yang ditentukan.

Proses *Bug Search and Fixing* dimulai dengan melakukan review kode sumber secara menyeluruh untuk mengidentifikasi potensi *bug* atau *area* yang memerlukan perbaikan. Tim magang menggunakan berbagai teknik dan alat analisis untuk membantu dalam proses ini, termasuk *debugging tools*, *static analysis tools*, dan *dynamic analysis tools*. Teknik-teknik ini membantu dalam mengisolasi *bug* dan memahami dampaknya terhadap performa dan fungsi aplikasi.

Setelah *bug* diidentifikasi, langkah selanjutnya adalah proses *fixing*. Ini melibatkan modifikasi kode sumber untuk memperbaiki *bug* dan memastikan bahwa aplikasi berfungsi dengan benar. Proses ini memerlukan pemahaman yang mendalam tentang struktur kode, logika program, dan bagaimana *bug* tersebut mempengaruhi perilaku aplikasi. Tim magang harus berhati-hati saat melakukan perubahan ini untuk tidak memperkenalkan *bug* baru atau mengganggu fungsi lain dalam aplikasi. Tahapan *Bug Search and Fixing* pada minggu ke-20 menunjukkan betapa pentingnya kualitas kode dalam pengembangan perangkat lunak. Melalui proses ini, tim magang belajar nilai dari detail, pentingnya dokumentasi, dan bagaimana komitmen terhadap kualitas dapat mempengaruhi keberhasilan proyek.

3.3 Kendala yang Ditemukan

Selama periode magang sebagai *full stack developer* di PT. Niagatama Poin Ciaprima, terdapat serangkaian tantangan yang cukup signifikan. Berikut beberapa tantangan yang terdapat pada magang :

1. Adanya keterbatasan waktu yang ketat pada proyek-proyek yang diberikan. Batas waktu yang ketat ini menimbulkan tekanan yang signifikan, karena hal tersebut harus berusaha menyelesaikan semua tugas dan pekerjaan yang ada dalam jangka waktu yang terbatas.
2. Terdapat kesulitan dalam menjaga keseimbangan keterampilan *front-end* dan *back-end* yang menjadi tantangan. Hal ini menjadi semakin rumit karena tugas-tugas yang berkaitan dengan *front-end* dan *back-end* diberikan kepada individu yang berbeda.
3. Mengimplementasikan desain responsif yang dapat beradaptasi dengan berbagai perangkat dan ukuran layar merupakan salah satu tantangan utama dalam pengembangan *front-end*. Hal ini menjadi semakin kompleks bagi individu yang masih dalam tahap pembelajaran awal dalam mengembangkan tata letak responsif.
4. Kesulitan dalam mendapatkan umpan balik langsung dan konstruktif dari mentor atau rekan kerja. Hal ini dapat memperlambat proses pembelajaran dan pertumbuhan profesional, karena umpan balik yang tepat waktu dan relevan sangat penting untuk mengidentifikasi area yang perlu ditingkatkan dan memahami bagaimana menerapkan keterampilan baru dengan efektif.

3.4 Solusi atas Kendala yang Ditemukan

Setelah memahami secara mendalam tantangan yang dihadapi, telah ditemukan beberapa solusi yang potensial untuk mengatasi kendala tersebut :

1. Dalam menghadapi tantangan keterbatasan waktu yang ketat pada proyek-proyek yang dihadapi, penting untuk menerapkan manajemen waktu yang efektif dan melakukan prioritas yang tepat terhadap pekerjaan yang harus selesai. Strategi ini mencakup pembagian tugas yang jelas, penggunaan alat-alat manajemen proyek yang efisien, serta komunikasi yang terbuka dan efektif dengan anggota tim. Melalui pendekatan ini, kita dapat lebih mudah mengidentifikasi tugas-tugas yang memiliki prioritas tertinggi dan memastikan bahwa sumber daya waktu yang ada digunakan dengan cara yang paling optimal.
2. Untuk mengatasi tantangan dalam mempertahankan keseimbangan antara keterampilan *front-end* dan *back-end*, strategi yang efektif adalah melakukan diskusi antar individu. Melalui pendekatan ini, anggota tim dapat memahami peran dan hambatan masing-masing bidang, sementara juga memperluas keterampilan mereka di kedua area tersebut. Hal ini akan menghasilkan kerja sama yang lebih efisien dalam pengembangan aplikasi web yang komprehensif.
3. Untuk mengatasi tantangan ini, salah satu pendekatan yang efektif adalah dengan memanfaatkan sumber daya yang menawarkan tutorial dan latihan praktis terkait pengembangan tata letak responsif. Dengan belajar secara mandiri dan melakukan praktik langsung, seseorang dapat mengasah pemahamannya tentang prinsip-prinsip desain responsif serta teknik-teknik yang diterapkannya. Selanjutnya, berinteraksi dan bekerja sama dengan komunitas pengembang lainnya, baik melalui platform *online* maupun komunitas lokal, dapat menjadi kunci dalam menyelesaikan masalah dan bertukar pengetahuan tentang metode terbaik dalam menghadapi hambatan desain responsif.

4. Mengadakan pertemuan reguler atau menyelenggarakan sesi pembinaan keterampilan secara teratur dapat memfasilitasi komunikasi yang lebih efisien. Selanjutnya, mengajukan permintaan umpan balik spesifik terkait proyek atau pekerjaan yang sedang dijalankan dapat memungkinkan penambahan informasi yang relevan. Mengedukasi lingkungan kerja yang mempromosikan partisipasi aktif dalam memberikan dan menerima umpan balik sangat berperan penting, sehingga setiap anggota tim merasa bebas untuk berbagi dan belajar satu sama lain dari pengalaman mereka.