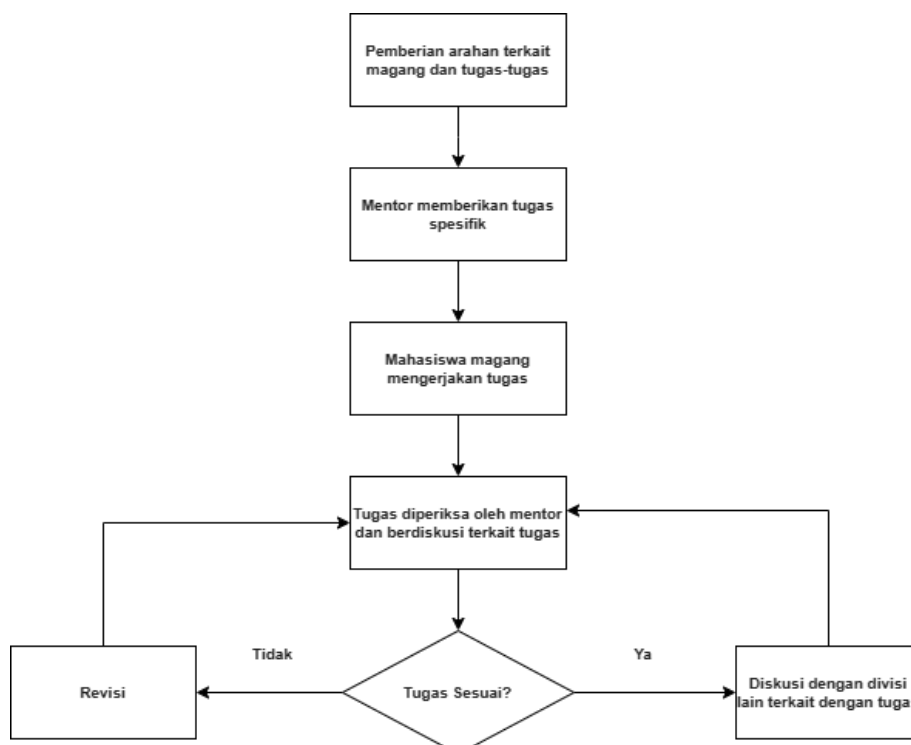


## BAB III

### 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[8].

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[9].

Dalam perancangan *e-commerce* untuk PT Nagatama Poin Ciaprima, tim mahasiswa magang menerapkan metode Pengembangan Sistem Life Cycle (SDLC). Ini mencakup tahap-tahap penting seperti analisis kebutuhan, desain sistem, pengembangan dan implementasi, pengujian dan validasi, serta *deployment* dan pemeliharaan. Pendekatan SDLC yang komprehensif memastikan bahwa aplikasi *e-commerce* yang dikembangkan memenuhi kebutuhan bisnis dan pengguna secara menyeluruh.

*Tabel 3. 1 Timeline Kerja Magang*

No	Tugas Yang Dilakukan	Tanggal Mulai	Tanggal Selesai
1-2	Pembuatan <i>Wireframe Website</i> Perusahaan	24 Januari 2024	26 Januari 2024
	Pembuatan <i>UI/UX Website</i> Perusahaan	29 Januari 2024	19 Maret 2024
3-4	Pembuatan <i>Front-end Website</i> Perusahaan	20 Maret 2024	5 April 2024
	Pembuatan <i>Back-end Website</i> Perusahaan	15 April 2024	3 Mei 2024
	Melakukan Integrasi antara <i>Front-end dan Back-end Website</i>	24 April 2024	03 Mei 2024

No	Tugas Yang Dilakukan	Tanggal Mulai	Tanggal Selesai
5	Testing	6 Mei 2024	13 Mei 2024
	<i>Bug Search and Fixing</i>	14 Mei 2024	25 Mei 2024

Pada laporan magang ini, tugas dan uraian kerja magang dikategorikan ke dalam *job description* yang tertera pada *Letter of Acceptance*. Aktivitas-aktivitas pada Tabel 3.1 yang dilakukan selama kerja magang adalah sebagai berikut:

**1. *Job Desc 1-2 (Mengembangkan, memelihara, dan meningkatkan teknologi Perusahaan): Pembuatan Wireframe Website Perusahaan***

Pada 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. Dalam tahap ini, mahasiswa magang bekerja sama dengan tim pengembang *website* untuk merancang *Wireframe* yang akan menjadi dasar bagi struktur dan tata letak halaman web. *Wireframe* adalah representasi visual sederhana dari antarmuka pengguna yang menggambarkan tata letak elemen-elemen kunci pada setiap halaman, seperti *header*, *footer*, navigasi, konten utama, dan fitur-fitur lainnya[10].

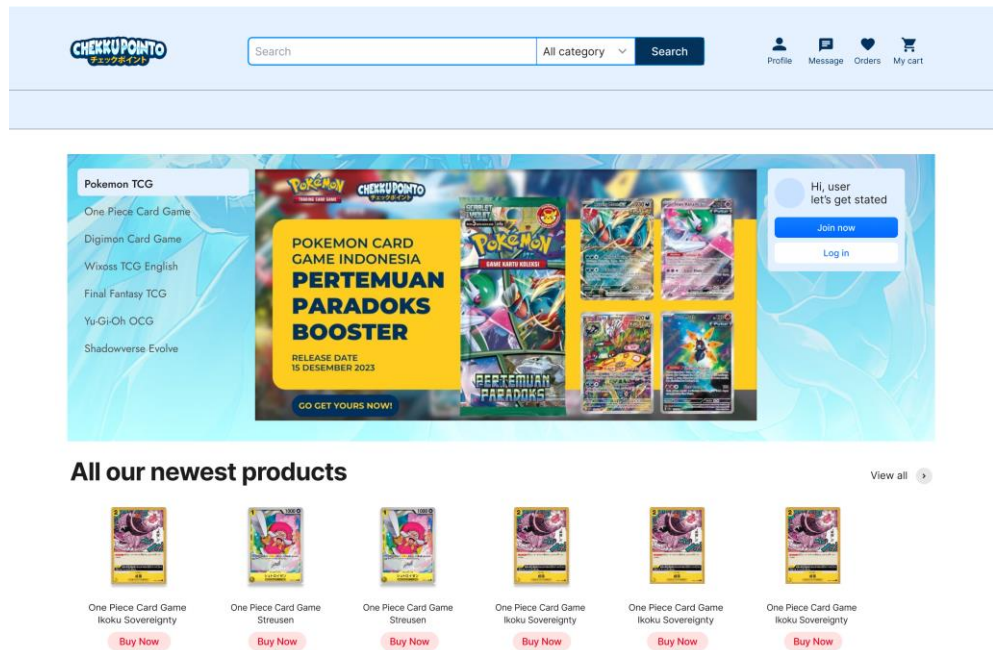
Selain itu, *wireframe* yang telah dibuat juga berfungsi sebagai alat bantu penting dalam merencanakan desain *e-commerce* yang akan dikembangkan di masa mendatang. Dengan memiliki kerangka dasar yang jelas, tim pengembang dapat memvisualisasikan struktur dan tata letak halaman web sebelum proses desain visual dimulai. Hal ini memungkinkan

mereka untuk mengidentifikasi potensi masalah, mengevaluasi efektivitas alur kerja, dan mengoptimalkan pengalaman pengguna sejak tahap awal pengembangan. Proses perancangan *Wireframe* ini merupakan langkah krusial dalam pengembangan *e-commerce* yang efektif. Dengan kolaborasi yang erat antara mahasiswa magang dan tim pengembang, mereka dapat menciptakan kerangka dasar yang kuat dan jelas, sehingga proses desain dan implementasi menjadi lebih efisien dan efektif. Hal ini tidak hanya memberikan pengalaman berharga bagi mahasiswa, tetapi juga memastikan bahwa *website* yang dikembangkan memenuhi kebutuhan pengguna dengan baik.

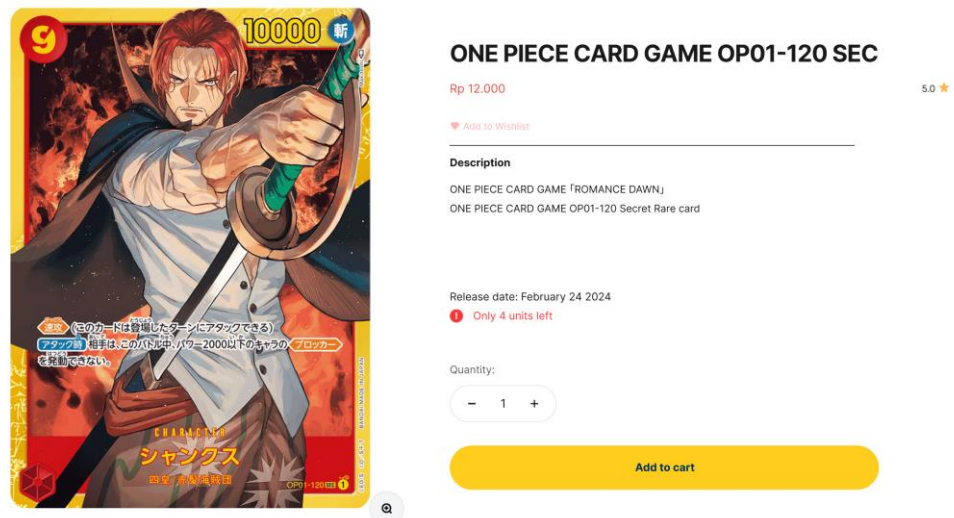
## **2. *Job Desc* 1-2 (Berkolaborasi dengan tim pengembang dan desainer untuk menciptakan solusi pengembangan yang baik) : Pembuatan *UI/UX Website Perusahaan***

Setelah menyelesaikan perancangan *Wireframe* pada minggu-minggu sebelumnya, para mahasiswa magang kini fokus pada pengembangan desain visual yang komprehensif untuk *website* perusahaan. Mereka bekerja erat dengan tim desain untuk mengembangkan konsep visual yang selaras dengan identitas *brand* dan memenuhi kebutuhan pengguna. 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. Dalam proses ini, diskusi yang berkelanjutan dengan tim *Project Management* menjadi sangat penting. Mahasiswa magang terlibat dalam pertemuan rutin untuk membahas kemajuan proyek, mengidentifikasi tantangan, dan memastikan bahwa semua elemen *website* dirancang dengan cara yang paling efisien dan efektif. 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. Melalui kolaborasi yang erat dengan berbagai divisi, para mahasiswa magang mengembangkan keterampilan komunikasi, pemecahan masalah, dan pemikiran strategis yang sangat berharga bagi karier mereka di masa depan.



Gambar 3. 2 Tampilan pada Figma



Gambar 3. 3 Tampilan pada Figma

Selama fase ini, tim magang menggunakan *software Figma* sebagai alat utama untuk merancang *User Interface (UI)* dan *User Experience (UX)*. *Figma* ini sendiri menghasilkan sebuah *prototype* seperti pada Gambar 3.2 dan 3.3 berdasarkan keputusan tim magang. 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. Berdasarkan desain *Figma* yang telah dikembangkan oleh para mahasiswa magang, *website* perusahaan akan dilengkapi dengan berbagai fitur yang dirancang untuk memberikan

pengalaman pengguna yang optimal. Halaman Beranda akan menampilkan *carousel* atau *slider* untuk mempromosikan produk dan layanan unggulan, serta informasi singkat tentang beberapa *event* pada perusahaan dan tombol aksi untuk mengakses informasi lebih lanjut. Halaman Produk akan menyajikan daftar produk atau layanan yang ditawarkan, lengkap dengan detail spesifikasi, harga, dan informasi lainnya. Fitur pencarian dan penyaringan akan memudahkan navigasi pengguna. Sementara itu, untuk memfasilitasi interaksi dengan pelanggan, *website* akan dilengkapi dengan Halaman Kontak yang menawarkan formulir kontak, informasi kontak, dan peta lokasi kantor. Selain itu, Area Pelanggan akan menyediakan fitur pemesanan *online*, pelacakan pesanan, dan akses ke riwayat pembelian.

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. Keseluruhan fitur yang dikembangkan oleh para mahasiswa magang telah memperhitungkan aspek kemudahan penggunaan, konsistensi visual, dan integrasi yang komprehensif. Implementasi fitur-fitur ini diharapkan dapat meningkatkan keterlibatan pelanggan, memperluas jangkauan pemasaran, dan meningkatkan efisiensi operasional perusahaan. Namun, tim pengembangan *website* harus menghadapi tantangan yang tak terduga. Adanya kendala terkait keuangan perusahaan menuntut penyesuaian dan perubahan pada rencana awal pengembangan tampilan antarmuka pengguna (UI). Situasi keuangan yang kurang menguntungkan ini membuat perusahaan harus berhati-hati dalam mengalokasikan anggaran untuk proyek ini. Oleh karena itu, tim pengembang harus beradaptasi dan mencari solusi desain UI yang lebih efisien, namun tetap fungsional dan memberikan pengalaman pengguna yang baik. Meskipun menghadapi tantangan terkait kendala keuangan, tim tetap berupaya untuk menghasilkan sebuah *website*



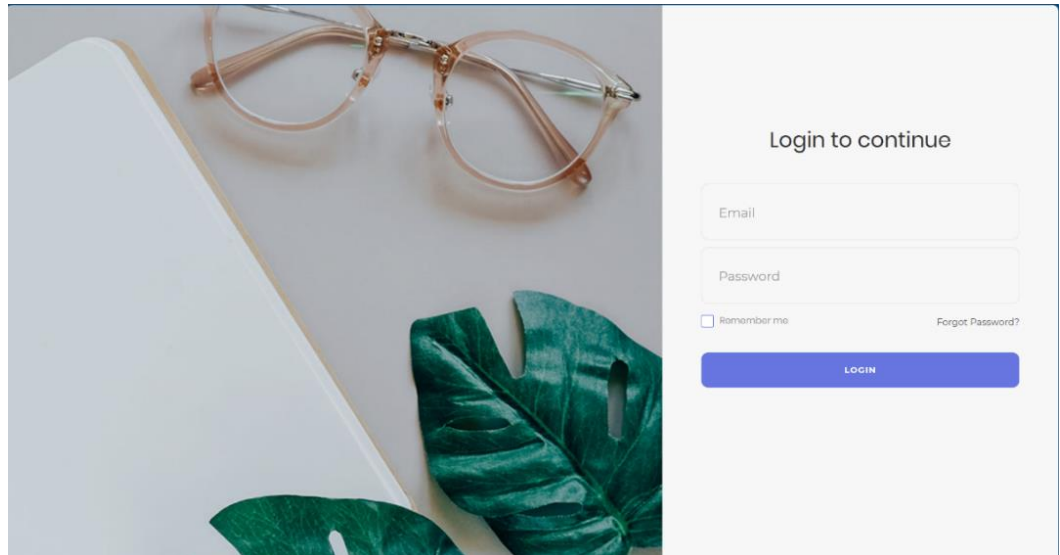
yang tidak hanya fungsional, tetapi juga dapat menarik minat dan memberikan pengalaman yang menyenangkan bagi pengguna. Fleksibilitas dalam beradaptasi dengan perubahan dan kemampuan mengoptimalkan sumber daya yang ada akan menjadi kunci keberhasilan dalam mengembangkan tampilan UI yang sesuai dengan anggaran yang tersedia.

### 3. *Job Desc 3-4 (Menangani pengembangan frontend dan backend): Pembuatan 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[11]. 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 Tampilan halaman login*

Pada Gambar 3.4 menunjukkan visualisasi dari 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. Halaman *login* berfungsi sebagai jembatan bagi setiap pengguna untuk dapat berinteraksi dengan platform digital. Pemahaman yang jelas mengenai peran dan posisi strategis halaman *login* ini akan membantu tim pengembangan untuk merancang tampilan dan alur interaksi yang optimal, sehingga dapat memberikan pengalaman pengguna yang baik dan mendukung tujuan keseluruhan dari proyek *website* ini. Tampilan sederhana dari halaman *login* yang ditunjukkan pada Gambar 3.4 telah didiskusikan sebelumnya oleh tim pengembangan. Desain ini dirancang untuk memenuhi kebutuhan dasar pengguna dalam mengakses *website*, dengan tetap mempertimbangkan faktor kemudahan penggunaan dan estetika yang sesuai dengan *brand* perusahaan. Kendala keuangan yang dihadapi saat ini

menjadi salah satu pertimbangan dalam pemilihan desain yang lebih sederhana, namun tetap fungsional dan efektif. Berikut merupakan *code* yang terdapat pada halaman *login* ini :

```

1 <html>
2 <head>
3 <title>chekku valetto official</title>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6
7 <!--
8 <link rel="icon" type="image/png" href="assets/icons/chekku.png"/>
9
10 <!--
11 <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
12
13 <!--
14 <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">
15
16 <!--
17 <link rel="stylesheet" type="text/css" href="fonts/linearicons-free-v1.0.0/icon-font.min.css">
18
19 <!--
20 <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
21
22 <!--
23 <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">
24
25 <!--
26 <link rel="stylesheet" type="text/css" href="vendor/magnific-popup/magnific-popup.css">
27
28 <!--
29 <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
30
31 <!--
32 <link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">
33
34 <!--
35 <link rel="stylesheet" type="text/css" href="css/util.css">
36 <link rel="stylesheet" type="text/css" href="css/main.css">
37
38 </head>
39 <body style="background-color: #e6e6e6;">
40
41 <div class="limiter">
42 <div class="container-login100">
43 <div class="wrap-login100">
44 <form class="login100-form validate-form" action="process/login.php" method="post">
45 <span class="login100-form-title p-b-4">
46 <input type="text" name="email" value="" />
47 </span>
48 <input type="password" name="pass" value="" />
49 <input type="checkbox" name="remember-me" />
50 <input type="submit" value="Login" />
51 </form>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>

```

Gambar 3. 5 Code CSS & Javascript login

```

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 CSS & Javascript login

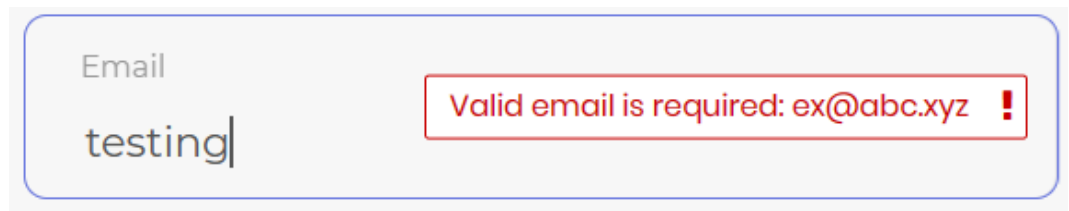
Code pada Gambar 3.5 dan 3.6 merupakan *code* untuk implementasi dari desain halaman *login* yang telah dijelaskan sebelumnya. Kode tersebut mengintegrasikan penggunaan CSS dan *JavaScript* dalam satu *project*, dengan tujuan untuk menciptakan tampilan *login* yang menarik dan interaktif bagi pengguna. Pada *code* terdapat beberapa *script JavaScript* yang berperan dalam meningkatkan pengalaman pengguna saat berinteraksi dengan halaman *login*. Skrip-skrip ini tidak hanya bertanggung jawab untuk menghasilkan tampilan yang menarik secara visual, namun juga memberikan validasi terhadap *input* yang diberikan oleh pengguna.

```
function validate (input) {
  if($(input).attr('type') == 'email' || $(input).attr('name') == 'email') {
    if($(input).val().trim().match(/^([a-zA-Z0-9_\.\-])+\@(([0-9]{1,3}\.){0-9}[1,3]\.){0-9}([0-9]{1,3}\.){0-9}([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 pemeriksaan terhadap input

Pada Gambar 3.7, merupakan sebuah *code javascript* yang disajikan berfungsi untuk melakukan pemeriksaan dan validasi terhadap *input* yang diberikan oleh pengguna, khususnya pada bagian email. Code ini merupakan bagian integral dari keseluruhan alur interaksi pada halaman *login*. Melalui *code* tersebut, sistem akan melakukan pengecekan apakah email yang dimasukkan oleh pengguna telah sesuai dengan format yang seharusnya. Validasi ini penting dilakukan untuk memastikan integritas data dan keamanan akun pengguna saat melakukan proses *login*.

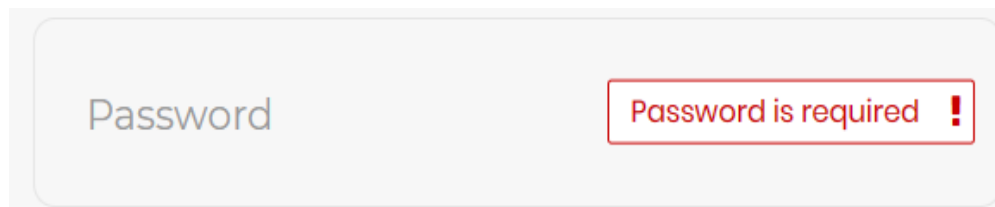
Apabila email yang dimasukkan tidak memenuhi format yang diharapkan, maka sistem akan memberikan peringatan atau notifikasi yang jelas kepada pengguna. Hal ini bertujuan untuk memandu pengguna dalam memasukkan kredensial yang benar, sehingga dapat melanjutkan proses *login* dengan lancar.



Gambar 3. 8 Peringatan jika input tidak sesuai

Berikut pada Gambar 3.8 menampilkan contoh peringatan (*warning*) yang akan muncul jika pengguna memasukkan *input* email yang tidak sesuai format seperti “*Valid email is required: ex@abc.xyz*”. Sebagaimana ditunjukkan pada potongan kode sebelumnya, sistem akan melakukan validasi terhadap email yang dimasukkan oleh pengguna. Apabila email yang dimasukkan tidak memenuhi format yang diharapkan, maka akan muncul peringatan visual yang akan memberi tahu pengguna tentang kesalahan tersebut. Dalam contoh yang ditampilkan, peringatan tersebut menjelaskan bahwa email yang dimasukkan tidak valid dan memberikan contoh format email yang benar.

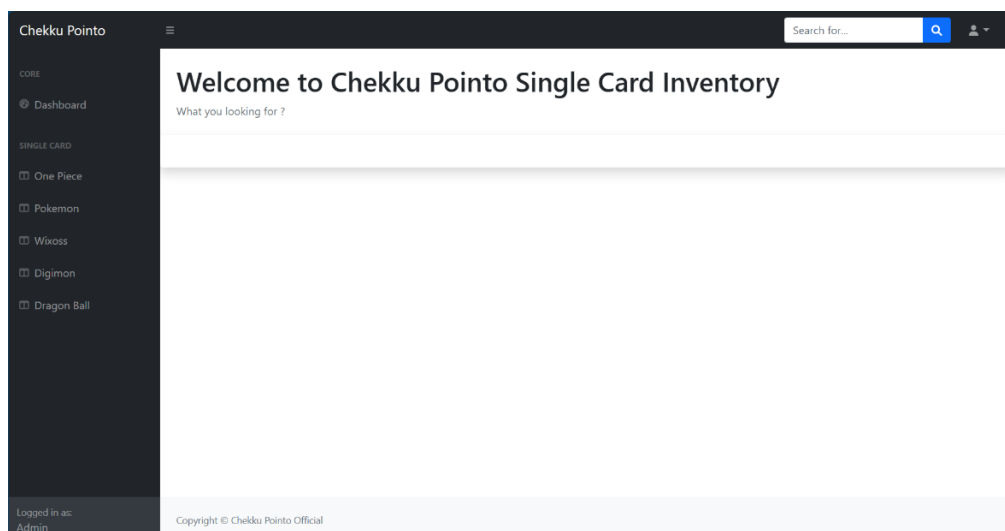
Fitur peringatan visual seperti ini sangat penting untuk memastikan pengalaman pengguna yang baik saat melakukan proses *login*. Dengan adanya informasi yang jelas dan contoh format yang benar, pengguna akan dapat dengan mudah memperbaiki *input* mereka dan melanjutkan proses *login* tanpa hambatan. Integrasi antara validasi email melalui skrip *JavaScript* dan peringatan visual yang informatif, seperti yang ditampilkan pada Gambar 3.8, menunjukkan upaya tim pengembangan untuk menciptakan halaman *login* yang *user-friendly* dan memastikan keamanan data pengguna.



Gambar 3. 9 Input tidak sesuai

Pada Gambar 3.9 menampilkan contoh peringatan (*warning*) yang akan muncul jika pengguna tidak memasukkan *password* saat melakukan *login*. Sama halnya dengan validasi email yang dijelaskan sebelumnya, sistem juga akan melakukan pemeriksaan terhadap *input password* yang diberikan oleh pengguna. Dalam contoh ini, apabila pengguna tidak memasukkan *password* sama sekali, maka akan muncul peringatan visual yang menyatakan "*Password is required*". Pesan peringatan ini bertujuan untuk memberitahu pengguna bahwa *input password* wajib diisi agar dapat melanjutkan proses *login*.

Integrasi fitur validasi *password* ini dengan peringatan visual yang informatif, seperti yang ditampilkan pada Gambar 3.9, merupakan upaya tim pengembangan untuk memastikan keamanan dan integritas akun pengguna. Fitur ini juga membantu dalam menciptakan pengalaman pengguna yang *user-friendly*, dengan memberikan panduan yang jelas saat terjadi kesalahan *input*. Dengan menerapkan validasi dan peringatan yang konsisten baik untuk *input* email maupun *password*, platform digital ini berusaha untuk membangun kepercayaan pengguna dan menjamin keamanan akses ke dalam sistem.



*Gambar 3. 10 Halaman utama dari website*

Pada Gambar 3.10 menampilkan halaman utama (*homepage*) dari *website* yang telah dirancang. Desain halaman ini mengusung tema minimalis dan modern, dengan tujuan untuk memudahkan pengguna dalam penggunaan *website*. Pendekatan minimalis dan modern pada desain *website* ini merupakan upaya untuk menciptakan pengalaman pengguna (*user experience*) yang optimal. Tampilan yang bersih, dengan penggunaan warna dan tata letak elemen yang seimbang, diharapkan dapat membuat pengguna merasa nyaman dan dapat dengan mudah menavigasi konten serta fitur-fitur yang tersedia.

Pendekatan yang berpusat pada pengguna (*user-centric*) ini menunjukkan komitmen tim pengembangan untuk menyediakan platform digital yang dapat dinikmati oleh semua orang. Upaya ini diharapkan dapat meningkatkan keterlibatan dan kepuasan pengguna saat berinteraksi dengan *website*.

```
<!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/simple-datatables@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">Logout</a></li>
          </ul>
        </li>
      </ul>
    </nav>
  </body>
</html>
```

Gambar 3.11 Code navigation bar

Pada Gambar 3.11 menampilkan potongan kode HTML dan CSS yang digunakan untuk membangun *navigation* bar pada halaman utama *website* yang terlihat di Gambar 3.10. Kode ini menunjukkan bahwa tim pengembangan telah memanfaatkan berbagai fungsi dan aturan CSS untuk membuat *navigation* bar yang menarik dan interaktif bagi pengguna. Penggunaan CSS dalam kode ini bertujuan untuk membuat tampilan

*navigation* bar yang lebih estetis dan responsif. Beberapa teknik CSS yang terlihat antara lain:

- a. Pengaturan posisi elemen menggunakan *position: fixed* untuk membuat *navigation* bar tetap terlihat saat pengguna menggulir halaman.
- b. Penerapan efek *hover* pada *link* navigasi dengan menggunakan aturan *:hover* untuk memberi *feedback* visual saat pengguna mengarahkan *pointer*.
- c. Pemanfaatan transformasi CSS seperti *transform: scale()* untuk memberikan efek pembesaran saat *link* navigasi *di-hover*.
- d. Penambahan transisi CSS *transition* untuk membuat efek animasi yang halus saat pengguna berinteraksi dengan *navigation* bar.

Integrasi kode HTML dan CSS yang terlihat pada Gambar 3.11 merupakan bagian penting dalam membangun halaman *website* yang *user-friendly* dan menarik bagi 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                         </a>
58
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                         </a>
63
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                         </a>
68
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                         </a>
73                     </div>
74                 </div>
75                 <div class="sb-sidenav-footer">
76                     <div class="small">Logged in as:</div>
77                     Admin
78                 </div>
79             </nav>
80         </div>
81     <div id="layoutSidenav_content">
82         <main>
83             <div class="container-fluid px-4">
84                 <h1 class="mt-4">Welcome to Chekku Pointo Single Card Inventory</h1>
85                 <ol class="breadcrumb mb-4">
86                     <li class="breadcrumb-item active">What you looking for ?</li>
87                 </ol>
88             </div>
89         </main>

```

Gambar 3. 12 Code untuk menu halaman utama

Pada Gambar 3.12 menampilkan potongan kode yang digunakan untuk membuat menu navigasi di sisi kiri halaman utama *website*, sebagaimana ditunjukkan pada Gambar 3.10. Tujuan dari menu ini adalah untuk memungkinkan pengguna berpindah ke halaman-halaman lain dalam *website*. Dalam pengembangan kode ini, tim telah memanfaatkan beberapa kelas (*class*) yang tersedia di *framework CSS* populer, yaitu *Bootstrap*. Penggunaan *Bootstrap* memudahkan mahasiswa magang yang terlibat dalam proyek ini untuk mengerjakan dan mengintegrasikan elemen-elemen antarmuka pengguna (UI) yang konsisten dan responsif. Pemanfaatan

*framework* CSS seperti *Bootstrap* menunjukkan upaya tim pengembangan untuk membangun *website* secara efisien dan menjaga konsistensi desain. Dengan menggunakan komponen-komponen yang sudah tersedia di *Bootstrap*, para pengembang dapat fokus pada logika fungsionalitas *website* tanpa harus membangun seluruh UI dari awal.

```
<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 menu pilihan pengguna

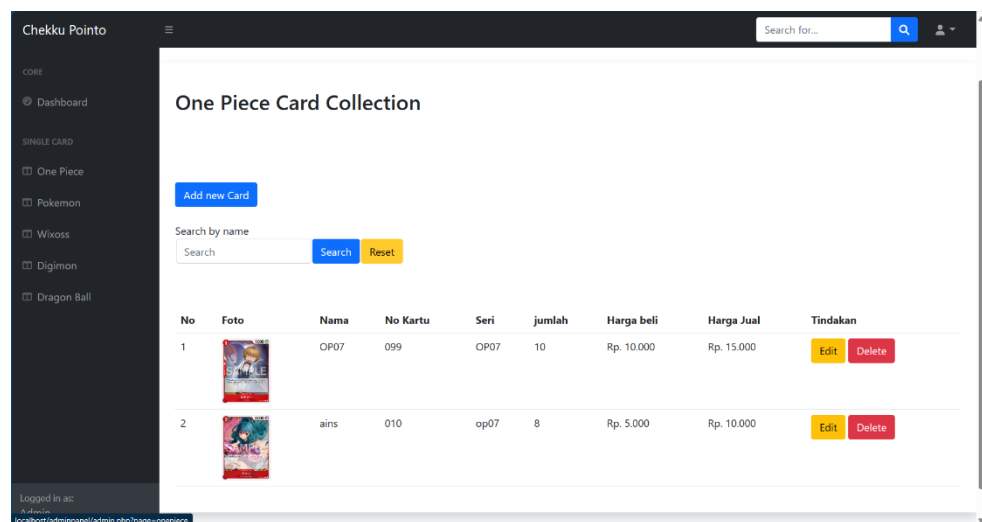
Pada Gambar 3.13 menampilkan potongan kode yang digunakan untuk mengimplementasikan fungsionalitas navigasi halaman pada *website*. Dalam kode tersebut, terlihat bahwa mahasiswa magang yang terlibat dalam pengembangan proyek ini memanfaatkan kemampuan PHP untuk mengelola perpindahan halaman sesuai dengan menu yang dipilih oleh pengguna.

Pendekatan yang digunakan adalah dengan menerapkan struktur kontrol *if-else* dalam kode PHP. Logika ini memungkinkan *website* untuk menampilkan halaman yang sesuai berdasarkan aksi pengguna, seperti saat

pengguna mengklik menu tertentu. Dengan memanfaatkan kemampuan pengolahan kondisi yang dimiliki oleh PHP, pengembang dapat dengan mudah menentukan halaman mana yang harus ditampilkan ketika pengguna berinteraksi dengan menu navigasi.

Penerapan pendekatan ini membawa beberapa keuntungan bagi *website* yang sedang dikembangkan. Pertama, *website* menjadi lebih efisien dalam hal navigasi dan pengalaman pengguna. Ketika pengguna berpindah dari satu halaman ke halaman lain, transisi tersebut dapat dilakukan dengan lancar dan responsif, tanpa harus memuat ulang seluruh halaman *website*.

Selain itu, dengan menerapkan logika navigasi berbasis PHP, pengembang *website* juga dapat dengan mudah menambahkan halaman-halaman baru ke dalam *website* di kemudian hari. Cukup dengan menambahkan kondisi baru dalam struktur *if-else*, pengembang dapat mengintegrasikan halaman baru tersebut ke dalam alur navigasi *website* tanpa harus melakukan banyak perubahan pada kode yang sudah ada.



Gambar 3. 14 Tampilan ketika pengguna memilih

Pada Gambar 3.14 menampilkan antarmuka halaman yang menampilkan produk-produk yang berkaitan dengan *One Piece*. Halaman ini menyediakan berbagai fitur interaktif yang memungkinkan pengguna untuk berinteraksi dengan konten yang ditampilkan. Ketika pengguna

masuk ke halaman ini, mereka akan disuguhkan dengan daftar produk yang terkait dengan seri *One Piece*. Tampilan ini memungkinkan pengguna untuk menelusuri berbagai macam barang, seperti *merchandise*, pakaian, atau bahkan salinan video atau buku dari *franchise* populer tersebut. Selain melihat daftar produk, pengguna juga dapat melakukan beberapa tindakan lain pada halaman ini. Mereka dapat menambahkan produk baru ke dalam daftar, misalnya jika mereka ingin menjual barang baru yang berkaitan dengan *One Piece*. Pengguna juga diberi kemampuan untuk mengubah informasi produk yang sudah ada, seperti harga atau stok barang.

Apabila terjadi kesalahan atau pengguna ingin menghapus suatu produk, halaman ini menyediakan opsi untuk menghapus item yang telah dimasukkan. Fitur ini memastikan bahwa pengguna dapat dengan mudah mengelola daftar produk sesuai kebutuhan mereka. Selain itu, halaman ini juga menampilkan sebuah kotak pencarian di bagian atas. Fitur ini memungkinkan pengguna untuk mencari produk tertentu yang mereka inginkan dengan cepat, tanpa harus menelusuri seluruh daftar produk yang ditampilkan. Jika pengguna ingin kembali ke tampilan awal setelah melakukan pencarian, terdapat tombol "*reset*" yang memungkinkan mereka untuk kembali ke daftar produk utama.

```

<?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>

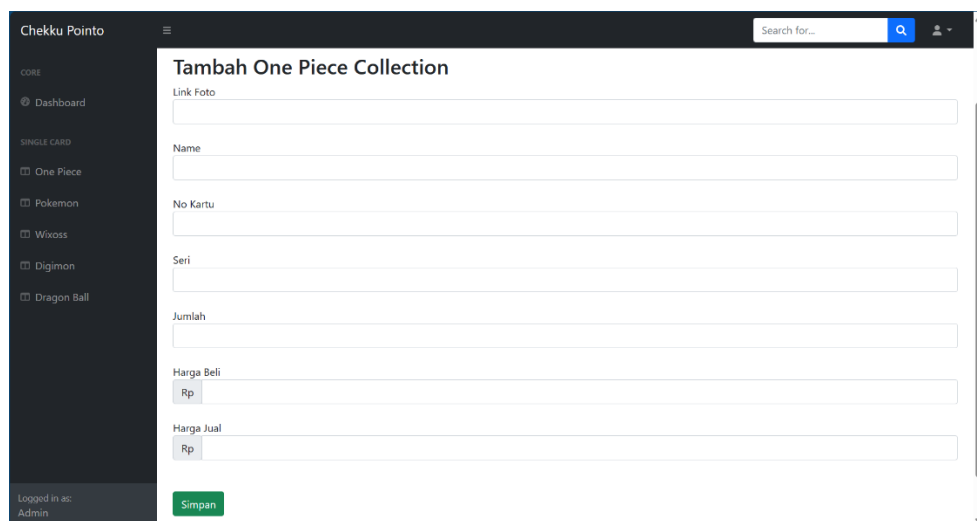
```

Gambar 3. 15 Code halaman website

Pada Gambar 3.15 menampilkan potongan kode yang digunakan untuk membangun antarmuka halaman yang ditunjukkan pada Gambar 3.14. Dalam kode ini, mahasiswa magang yang terlibat dalam pengembangan proyek ini memanfaatkan fungsi tabel untuk mempermudah pengguna dalam melihat dan berinteraksi dengan daftar produk yang tersedia. Penggunaan struktur tabel dalam kode ini memberikan beberapa keuntungan bagi pengguna. Pertama, format tabel memungkinkan

penyajian informasi produk secara terstruktur dan rapi, sehingga pengguna dapat dengan mudah menelusuri dan memahami daftar produk yang ditampilkan. Baris-baris dalam tabel dapat digunakan untuk mewakili setiap produk individual, sementara kolom-kolom dapat dimanfaatkan untuk menampilkan detail-detail penting dari masing-masing produk, seperti nama, harga, stok, atau informasi lainnya.

Selain itu, penggunaan tabel juga memudahkan pengguna dalam melakukan interaksi dengan daftar produk. Misalnya, pengguna dapat dengan mudah menambahkan produk baru dengan mengisi baris kosong dalam tabel, atau mengubah informasi produk yang sudah ada dengan menyunting data pada sel-sel terkait. Fitur penghapusan produk juga dapat diimplementasikan dengan lebih efektif melalui penggunaan struktur tabel.



*Gambar 3. 16 Tampilan ketika menambahkan product*

Pada Gambar 3.16 menampilkan antarmuka yang disediakan untuk pengguna *website* ketika mereka ingin menambahkan produk baru yang berkaitan dengan serial *One Piece*. Pada halaman ini, telah tersedia berbagai formulir yang diperlukan untuk melengkapi informasi produk yang akan ditambahkan.

Perancangan antarmuka ini menunjukkan upaya mahasiswa magang untuk menciptakan pengalaman pengguna yang mudah dipahami dan dioperasikan. Pada bagian ini elemen-elemen penting yang harus disediakan, seperti bidang *input* untuk nama produk, deskripsi, harga, stok, dan sebagainya. Dengan menyusun formulir-formulir tersebut dalam tata letak yang jelas dan intuitif, pengguna dapat dengan mudah memahami apa yang perlu mereka lakukan untuk menambahkan produk baru. Selain itu, mahasiswa magang juga mempertimbangkan aspek teknis dalam perancangan halaman ini. Salah satunya adalah terkait dengan format unggahan foto produk. Dari informasi yang tersedia, tampaknya perusahaan tempat mahasiswa magang bekerja telah terbiasa menyimpan foto-foto produk di situs web. Oleh karena itu, mereka memutuskan untuk menyediakan bidang *input* berbentuk tautan (*link*) untuk foto, sehingga memudahkan pengembang dalam mengunggah dan menampilkan foto produk di halaman *website*.

```

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

```

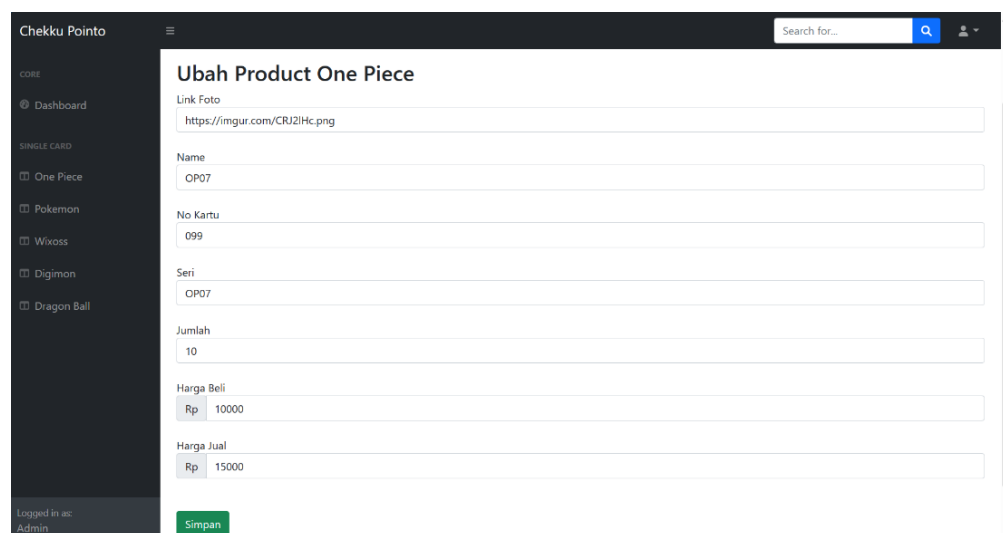
Gambar 3. 17 Code untuk menampilkan halaman

Pada Gambar 3.17 menampilkan *code* yang digunakan untuk membangun antarmuka halaman penambahan produk yang ditunjukkan pada Gambar 3.16. Namun, *code* ini tidak hanya digunakan untuk halaman penambahan produk, tetapi juga dimanfaatkan untuk halaman perubahan produk. Ini merupakan sebuah pendekatan yang cerdas dan efisien oleh mahasiswa magang dalam pengembangan situs web. Dengan menggunakan satu set kode yang dapat diterapkan di berbagai halaman yang membutuhkan fungsi serupa, seperti penambahan dan perubahan produk,



mahasiswa magang telah berhasil menerapkan prinsip-prinsip pengembangan perangkat lunak yang efektif. Pendekatan ini tidak hanya menghemat waktu dan upaya pengembangan, tetapi juga menjamin konsistensi dan kemudahan pemeliharaan kode di seluruh aplikasi.

Ketika pengguna ingin menambahkan produk baru, kode ini akan menyediakan antarmuka yang sudah *familiar* dan intuitif, sesuai dengan rancangan halaman yang ditunjukkan pada Gambar 3.16. Namun, ketika pengguna perlu melakukan perubahan pada produk yang sudah ada, kode yang sama dapat digunakan untuk menyediakan antarmuka yang serupa, sehingga pengguna dapat dengan mudah memahami dan melakukan proses perubahan.



*Gambar 3. 18 Tampilan untuk merubah product*

Pada Gambar 3.18 yang ditampilkan, terlihat antarmuka untuk melakukan perubahan terhadap produk tertentu. Halaman ini dirancang khusus untuk memungkinkan pengguna melihat dan mengubah nilai-nilai dari produk yang akan diperbarui. Berbeda dengan halaman penambahan produk baru, halaman ini mengambil nilai-nilai produk yang akan dimodifikasi dari basis data untuk ditampilkan di dalam formulir.

Proses pengambilan data produk dari basis data dilakukan melalui penggunaan fungsi SELECT pada bahasa pemrograman PHP. Fungsi ini bertugas menarik informasi terkait produk yang dipilih dari tabel dalam basis data, lalu menampilkannya di halaman web agar pengguna dapat melihat dan melakukan perubahan sesuai kebutuhan. Meskipun terdapat kesamaan dalam konsep dasar antara halaman penambahan produk (Gambar 3.17) dan halaman perubahan produk (Gambar 3.18), susunan kode yang digunakan berbeda. Hal ini wajar mengingat meskipun fungsionalitasnya serupa, terdapat kebutuhan untuk menyesuaikan alur dan tampilan antarmuka berdasarkan konteks penggunaannya.

Pada halaman perubahan produk ini, kode harus menyediakan mekanisme untuk mengambil nilai-nilai produk yang ada dari basis data dan menampilkannya di dalam formulir. Ini berbeda dengan halaman penambahan produk yang cenderung fokus pada pengumpulan *input* baru dari pengguna. Dengan memisahkan logika dan tampilan antara kedua halaman ini, pengembang dapat memastikan kode yang digunakan tetap terorganisir, mudah dipelihara, dan dapat digunakan kembali di tempat lain jika diperlukan.

4. *Job Desc 3-4 (Menangani pengembangan frontend dan backend):*  
**Pembuatan Back-end Website Perusahaan**

```
1 <?php
2
3 function connection(){
4     try{
5         $dsn = "mysql:host=localhost;dbname=chekcutest";
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 menghubungkan website

Pada Gambar 3.19 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. Dalam proyek ini, tim menggunakan *database dummy*, yang berarti *database* tersebut hanya digunakan untuk tujuan pengembangan dan pengujian tanpa mengubah *database* yang sudah ada sebelumnya. Pendekatan ini sangat bijaksana karena memungkinkan tim untuk melakukan inovasi dan eksperimen tanpa berdampak pada data yang sedang digunakan dalam lingkungan produksi.

Pada kode tersebut, mahasiswa magang menggunakan fungsi PDO (PHP Data *Objects*), yang merupakan sebuah antarmuka yang disediakan oleh bahasa pemrograman PHP. PDO menyediakan sebuah cara yang konsisten untuk berinteraksi dengan berbagai jenis *database*, mulai dari *database* relasional seperti *MySQL*, *PostgreSQL*, *Oracle*,

hingga *database NoSQL*. Penggunaan PDO memberikan beberapa keuntungan penting bagi tim pengembangan.

Pertama, PDO memungkinkan kode *database* menjadi lebih portabel. Tim dapat dengan mudah memindahkan kode dari satu *database* ke *database* lain tanpa perlu mengubah banyak bagian dari kode tersebut. Hal ini sangat berguna ketika proyek harus beradaptasi dengan perubahan kebutuhan atau migrasi *database* di kemudian hari. Selain itu, PDO juga menyediakan fitur-fitur keamanan yang membantu mencegah serangan *SQL injection*, salah satu ancaman keamanan yang umum terjadi pada aplikasi web. Penggunaan PDO dapat membantu menjaga keamanan aplikasi dengan cara yang lebih aman dibandingkan dengan menggunakan metode koneksi *database* yang lebih sederhana.

```

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

```

Gambar 3. 20 Code melakukan proses login

Pada Gambar 3.20 terdapat *code* untuk melakukan sebuah proses *login* yang menghubungkan antara pengguna dan halaman *website*. Dalam proses *login* ini, mahasiswa magang menggunakan bahasa pemrograman *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 di *input*

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 dilakukan 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 di *input* 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 untuk membuka page

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 pencarian terhadap product

Pada Gambar 3.22, sebuah *code* untuk melakukan pencarian 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.CategoryID FROM product p
        JOIN category c ON p.CategoryID = p.CategoryID
        Where p.CategoryID = 1"
        ;
$result = $pdo->prepare($sql);
$result->execute();
$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. 23 Code jika pengguna tidak ingin melakukan sebuah pencarian

Pada Gambar 3.23, merupakan sebuah *code* jika pengguna tidak ingin melakukan sebuah pencarian. Dengan *code* ini akan menampilkan segala data yang berhubungan dengan *one piece* di dalamnya. 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 untuk halaman formulir

Pada Gambar 3.24 merupakan sebuah *code PHP* untuk halaman formulir yang digunakan dalam menambah maupun melakukan perubahan terhadap nilai yang berada di dalam *database*. *Code PHP* yang terdapat pada Gambar 3.23, 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 yang memiliki fungsi untuk melakukan pengisian nilai

Pada Gambar 3. 25 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 didapatkan dari *database* akan di pindahkan ke dalam 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 front-end yang diintegrasikan

Pada Gambar 3.26 merupakan *code front-end* yang telah diintegrasikan dengan *PHP* yang berfungsi untuk mengisi nilai-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';
$db = connection();

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

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

    header("Location: ../admin.php?page=manage");
}

```

Gambar 3. 27 Code untuk menambah product

Pada Gambar 3.27 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 untuk melakukan perubahan

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 diubah 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 memasukkan 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 untuk penghapusan

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* ke dalam *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 di dalam *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.

##### 5. **Job Desc 5 (Menjamin keamanan dan kinerja sistem): 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[12].

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[13].

#### 6. **Job Desc 5 (Menjamin keamanan dan kinerja sistem): *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. Nagatama 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.