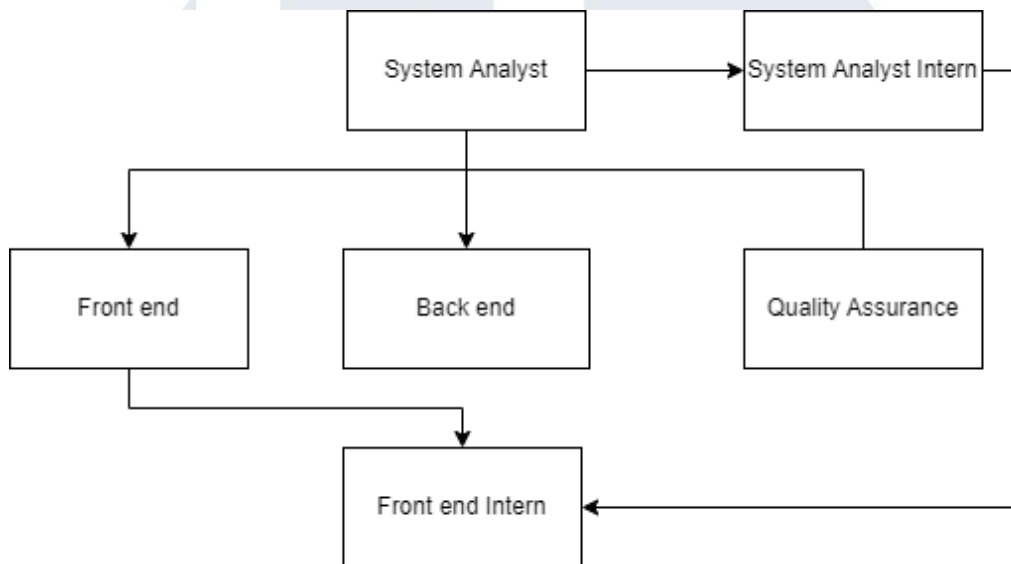


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



Gambar 3. 1 Kedudukan dan koordinasi

Dalam melaksanakan magang pada PT Ritz Project visi utama penulis memiliki dua kedudukan yaitu sebagai *system analyst* dan *Front-end developer*. Pelaksanaan magang dilakukan secara *Work from Home (WFH)* dengan sistem monitoring. Oleh karena itu, Bapak Farid ananda dengan jabatan *Project Manager Officer (PMO)* melakukan bimbingan dan *monitoring* selama pelaksanaan magang. Pada saat mengerjakan project magang, penulis di bantu oleh Pak Rizal selaku tim dari PT Ritz Project. dalam pelaksanaan magang dua orang pada posisi *system analyst*, dua orang pada posisi *front end developer*, 4 orang pada posisi *back-end developer*, dan satu orang pada posisi *Quality Assurance*, dengan total 8 orang dalam satu tim. Posisi *System analyst* melakukan koordinasi dengan tim *front-end* dan juga *back-end*, sedangkan *Front-end developer* melakukan koordinasi dengan tim *back-end developer*. Pada akhir magang penulis akan memberikan bantuan kerja kepada divisi *Quality Assurance* untuk membuat skenario *testing* dan melakukan manual *testing*.

Tugas yang dijalankan oleh *system analyst* adalah membuat segala jenis dokumentasi untuk *project*. Dalam kedudukan sebagai *system analyst* penulis berfokus pada pembuatan mockup dengan menggunakan *tools* Figma. *System analyst* harus mampu untuk melakukan koordinasi dengan divisi lain dalam *project* ini,

Peran sebagai *front-end developer* memiliki tugas untuk melakukan implementasi design *mockup* ke dalam bentuk *website*. Pembuatan tampilan *website* menggunakan bahasa pemrograman *html*. Untuk membuat tampilan menjadi *responsive* penulis menggunakan framework *Bootstrap 5*.

3.2 Tugas dan Uraian Kerja Magang

Dalam pelaksanaan kerja magang, penulis akan mengerjakan *project* untuk mengembangkan sistem *e-commerce* berbasis *website*. kerja magang berlangsung dari tanggal 26 Juni 2023 hingga 25 November 2023. Terdapat alur kerja yang diikuti selama progres kerja magang. Segala progres yang dilakukan dalam tim harus melalui persetujuan dari *Project Manager Officer (PMO)* Ritz Project. berikut adalah tugas yang telah di kerjakan oleh penulis:

Tabel 3. 1 Tugas kerja magang

No	Pekerjaan	Minggu	Tanggal Mulai	Tanggal Selesai
1	Pengenalan diri kepada perusahaan, briefing tugas yang akan dikerjakan selama magang, dan <i>handover</i> project	1	26 Juni 2023	30 Juni 2023
2	Pembuatan flowchart, mockup <i>UI/UX</i> login admin, user, dan register admin pada figma	2	3 Juli 2023	8 Juli 2023
3	Pembuatan front end <i>login</i> user, admin dan register admin	3	10 Juli 2023	14 Juli 2023

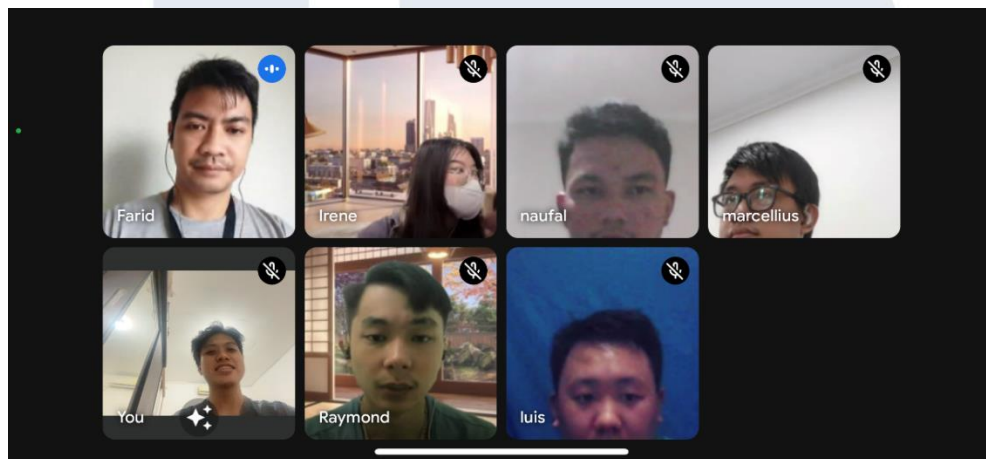
4	Update mockup UI/UX card item, dan Pembuatan mockup master event	4	17 Juli 2023	21 Juli 2023
5	Pengerjaan bagian pemisahan hak akses <i>user</i> , <i>admin</i> , dan super admin Pembuatan front end master event	5	24 Juli 2023	29 Juli 2023
6	Pembuatan activity diagram master diskon, Mockup master diskon, dan Front end master diskon	6	31 Juli 2023	5 Agustus 2023
7	Pembuatan flowchart, activity diagram, Mockup, dan front end fitur pay later	7	7 Agustus 2023	12 Agustus 2023
8	Pengerjaan Mockup dan Front end Info poin, Master Operational Hour, Master Holiday, Update tampilan detail item.	8-9	14 Agustus 2023	26 Agustus 2023
9	Pembuatan mockup, front end penempatan informasi operational hour, dan Pengerjaan Front end Master Sales	10	28 Agustus 2023	2 September 2023
10	Mempelajari slier item, dan pembuatan slider item pada tampilan <i>home page</i>	11	4 September 2023	9 September 2023
11	revisi slider card pada tampilan <i>home page</i>	12	11 September 2023	16 September 2023
12	Pembuatan mockup fitur retur sisi user dan admin menggunakan figma	13	18 September 2023	23 September 2023

13	Pembuatan front end tampilan fitur retur sisi user dan admin	14	25 September 2023	30 September 2023
14	Coding penambahan back button	15	2 Oktober 2023	7 Oktober 2023
15	Membantu membuat skenario testing modul user return, payterm, dan modul user transaction detail	16-17	9 Oktober 2023	21 Oktober 2023
16	Membantu manual testing modul user return, payterm, dan modul user transaction detail	18-19	23 Oktober 2023	4 November 2023
17	Fixing tampilan fail testing	20	6 November 2023	11 November 2023
18	<i>Phase Feedback UAT Result</i>	21-22	13 November 2023	25 November 2023

1. Pengenalan diri kepada perusahaan, briefing tugas yang akan dikerjakan selama magang, dan *handover* project pada tanggal 26 Juni 2023 – 30 Juni 2023

Pada minggu pertama penulis melakukan perkenalan kepada Bapak Farid, selaku mentor yang akan membimbing selama magang berlangsung. Penulis juga melakukan perkenalan kepada anggota tim magang, dalam tim magang terdapat 8 anggota. Mentor menjelaskan tugas project yang akan dilaksanakan selama magang serta alur kerja SLDC. Penulis mendapatkan pengetahuan mengenai bagaimana alur kerja SLDC seperti perencanaan, desain, implementasi, pengujian.

Pertemuan dilakukan secara online menggunakan Google meet yang dilaksanakan selama kurang lebih dua jam, di adakan meet agar para setiap anggota serta mentor untuk meningkatkan hubungan. Selama pertemuan berlangsung juga membahas mengenai latar belakang dari masing masing anggota serta mentor.

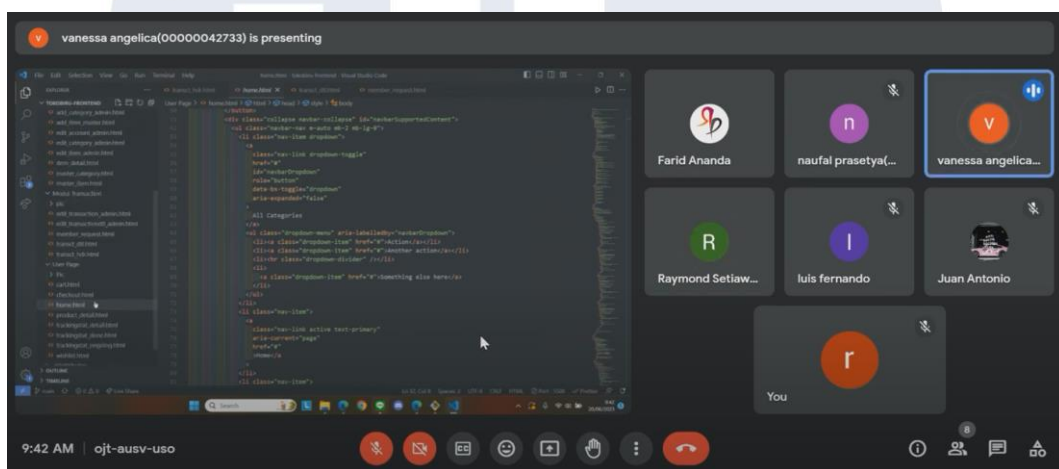


Gambar 3. 2 Pertemuan pertama

Pada minggu pertama penulis dan tim melakukan pertemuan dengan tim pemegang pada *batch* pertama. Pertemuan ini dilaksanakan untuk melakukan serah terima *project* website. mereka menjelaskan *project* apa saja yang sudah dikerjakan serta kekurangan yang masih terdapat pada *project*. Serah terima *project* dilaksanakan pada setiap divisi, dalam pertemuan dilangsungkan juga bedah *project* dimana tim *batch* pertama menjelaskan secara detail seluruh dokumen, *coding*, serta list *testing*.

Pada serah terima divisi *system analyst* pemegang *batch* pertama memberikan seluruh *project* yang telah di buat pada divisi ini. *Project* yang diberikan pada divisi *system analyst* yaitu dokumentasi spesifikasi program (SP) dan *database* website. Pada tahap ini pemegang *batch* pertama menjelaskan isi dari dokumen seperti *flowchart*, *activiyy diagram*, dan lain lain. Dokumentasi dibuat menggunakan google docs. Pemegang *batch* pertama menjelaskan fungsi dan alur tabel pada *database*, pembuatan *database* menggunakan aplikasi Dbvear.

Pada serah terima divisi *front-end batch* pertama menjelaskan seluruh project yang di berikan, project yang di berikan berupa file dengan format html. File yang diberikan merupakan tampilan website project. file di berikan melalui github, file project dapat dijalankan menggunakan aplikasi visual studio code. Agar tampilan menjadi lebih *responsive* menggunakan framework bootstrap 5. Bootstrap adalah framework yang berfungsi agar membuat tampilan website menjadi *responsive* dengan menggunakan html, css, dan javascript.



Gambar 3. 3 Handover Project

Setelah memahami seluruh project yang telah di berikan oleh batch pertama penulis mengunduh seluruh file project yang telah diberikan. tahapan selanjutnya adalah mengunduh seluruh aplikasi yang digunakan untuk dapat menjalankan seluruh file yang di berikan. Berikut adalah aplikasi yang di butuhkan untuk mendukung jalannya proses kerja magang:

Tabel 3. 2 Aplikasi yang dibutuhkan untuk magang

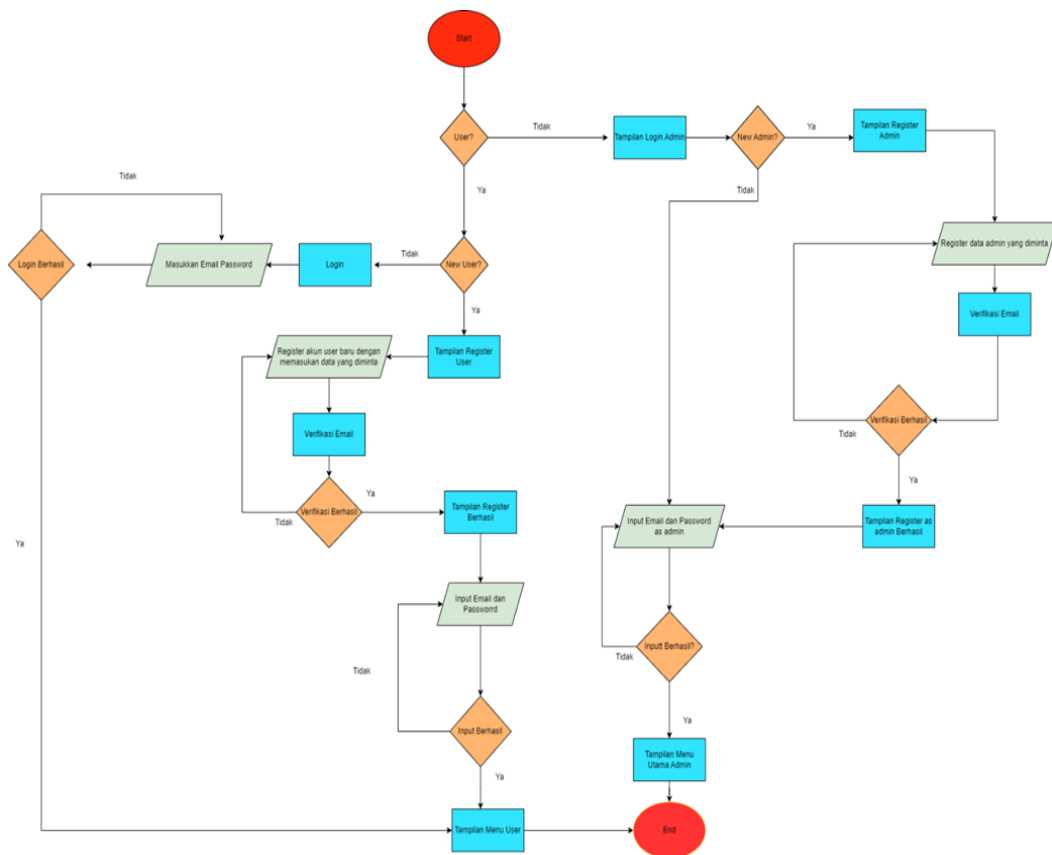
No	Tools	Function
1	Visual Studio Code	Editor
2	DBeaver	Database Mysql

3	Xampp	Web Server
4	Google Document	Report
5	Figma	Mockup
6	Katalon	Testing

2. 3 Juli 2023 – 8 Juli 2023

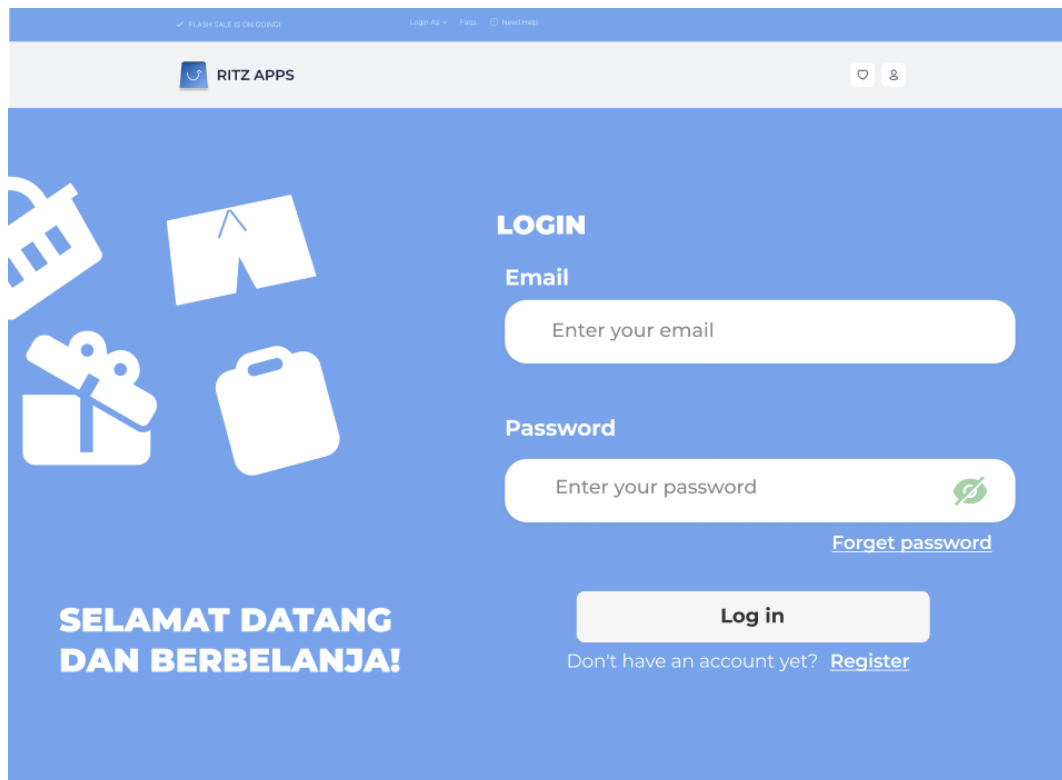
Pada minggu kedua, tim dan supervisor mulai membahas mengenai pengembangan yang akan dilakukan pada website ecommerce. Pada tahap ini, pengembahasan awal yang akan dilakukan adalah merubah alur penggunaan website dan konsep penjualan pada website. awalnya konsep penggunaan website hanya terdapat satu toko yang dapat menjual barang. Sehingga, transaksi penjualan hanya terjadi antara pembeli dan satu toko saja. Untuk mempermudah dan memberikan gambaran arah pengembangan website ecommerce tim penulis menggunakan ecommerce tokopedia sebagai referensi. Pengembangan yang akan dilakukan adalah dalam satu website ecommerce akan terdapat lebih dari satu toko yang dapat melakukan penjualan. Dalam tahapan ini, website ecommerce akan terbagi menjadi dua role saat *login* yaitu sebagai user/pengguna dan sebagai admin/pemilik toko. Berikut alur *login* dengan role *admin* dan *user*:

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3. 4 Flowchart login user dan admin

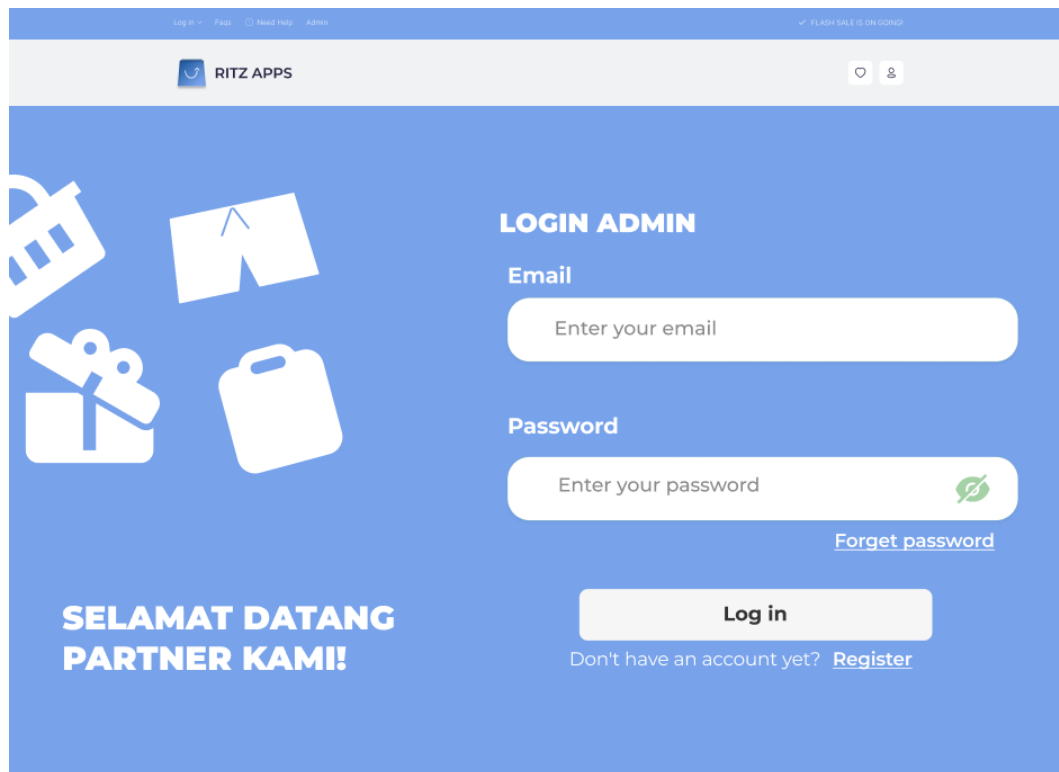
Pada Gambar 3.4 merupakan *flowchart login* website ecommerce, tampilan website akan di mulai dari halaman *login*, pada tampilan *login* akan di arahkan kepada dua pilihan antara login sebagai user atau admin. Jika login sebagai user namun belum memiliki akun akan masuk ke dalam halaman registrasi, apabila sudah memiliki akun bisa langsung login dengan akun yang telah di miliki dan langsung masuk ke dalam tampilan homepage website. alur *login* dengan *admin* memiliki alur yang serupa, jika belum memiliki akun *admin* maka akan diarahkan untuk melakukan registrasi, apabila sudah memiliki akun bisa *login* dan akan masuk ke dalam tampilan admin.



Gambar 3. 5 Mockup login user

Pada gambar 3.5 merupakan mockup halaman login sebagai user. Pada bagian navbar terdapat combo box yang berguna untuk memilih login sebagai user atau admin, pada bagian kiri atas terdapat logo dari website ecommerce ritz project. pada bagian isi halaman terbagi menjadi dua. pada sebelah kiri terdapat ucapan selamat datang “Selamat datang dan berbelanja!”. Pada sebelah kanan terdapat text box untuk memasukan email dan juga password, terdapat button “Login”. apabila user belum memiliki akun pengguna bisa menekan kata “register” yang terhubung dengan halaman register.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 6 Mockup login admin

Pada gambar 3.6 diatas merupakan mockup tampilan login untuk admin. Tampilan tidak banyak berbeda dari halaman login. Pada bagian atas terdapat navbar yang berisi *combo box* untuk memilih login sebagai user atau admin, terdapat logo dari website ecommerce pada kiri atas halaman. Pada bagian tengah halaman terbagi menjadi dua bagian. Pada sebelah kiri terdapat ucapan selamat datang “Selamat datang partner kami”. Pada sebelah kanan terdapat text box email dan password, terdapat button “login” untuk melakukan login, dibawahnya terdapat kata “register” untuk melakukan registrasi apabila belum memiliki akun.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 7 Mockup register admin

Pada gambar 3.7 merupakan tampilan untuk registrasi admin. Pada bagian atas terdapat navbar yang berisi combo box untuk memilih login sebagai admin atau user, pada kiri atas halaman terdapat logo website ecommerce. Pada bagian utama halaman register terbagi menjadi dua bagian. Pada bagian kiri terdapat gambar sebagai estetika halaman, lalu dibawahnya terdapat ucapan selamat datang “Selamat datang partner kami!”. Pada bagian kanan terdapat *text box* untuk *input* store name, email, phone number, address, dan password. Pada bagian bawah terdapat button “register”.

3. 10 Juli 2023 – 14 Juli 2023

Pada minggu ke tiga, penulis membuat tampilan front end login user, admin, dan register admin. Pembuatan *front end* mengikuti *mockup* UI/UX website yang telah dibuat. Pembuatan front end menggunakan HTML, CSS, dan Javascript pada aplikasi visual studio code ditambah dengan *framework* Bootstrap 5.

```

!-- Required meta tags -->
meta charset="utf-8">
meta name="viewport" content="width=device-width, initial-scale=1">

!-- Bootstrap CSS -->
link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integ

!-- Import font -->
link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

!-- Import font awesome -->
link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.3.0/css/all.min.css">

```

Gambar 3. 8 Coding import

Pada gambar 3.8 di atas merupakan bagian awal code front end untuk tampilan *login user, admin*, dan register admin. Pada gambar di atas terdapat code untuk import Bootstrap 5 yang fungsinya untuk membuat tampilan website menjadi semakin responsif, import font dengan jenis font yang di gunakan adalah Family Montserrat, import font awesome yang berguna untuk memberikan opsi untuk pemilihan ikon.

```

<!-- Navbar -->
<section id="navbar">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <div class="col justify-content-center text-center">
        <a class="text-decoration-none small text-light me-3" href="#"><i class="fa-solid fa-check
        <div class="btn-group">
          <button class="btn btn-sm dropdown-toggle text-light small" type="button" data-bs-toggle=
            LogIn AS
          </button>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item small" href="#">User</a></li>
            <li><a class="dropdown-item small" href="#">Admin</a></li>
          </ul>
          <a class="nav-link small text-light">Faq</a>
          <a class="nav-link small text-light">Need Help<i class="fa-solid fa-exclamation ms-1"></i>
        </div>
      </div>
    </nav>

    <nav class="navbar navbar-expand-lg navbar-light bg-light p-3">
      <div class="container-fluid">
        
        <a class="navbar-brand ms-1 fw-bolder" href="#">RITZ APPS</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbart
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav ms-auto">

```

Gambar 3. 9 Coding navbar front end

Pada gambar 3.9 diatas merupakan codingan untuk *navbar* yang akan di gunakan pada halaman login user, admin, dan register admin. Pada *navbar* terdapat

button combo box bertuliskan *LogIn As* dengan bentuk *drop down*. Terdapat dua pilihan yaitu *user* dan *admin*. Disamping *combo box* terdapat dua label yang dapat di klik dengan tulisan *FAQ*, dan *Need help*. Dibagian kiri atas terdapat logo dari website *ecommerce* dan terdapat label nama dari website *ecommerce*. Logo dari website memiliki tipe sebagai *button*, sehingga ketika logo diklik maka akan mengarah ke tampilan yang sudah dihubungkan.

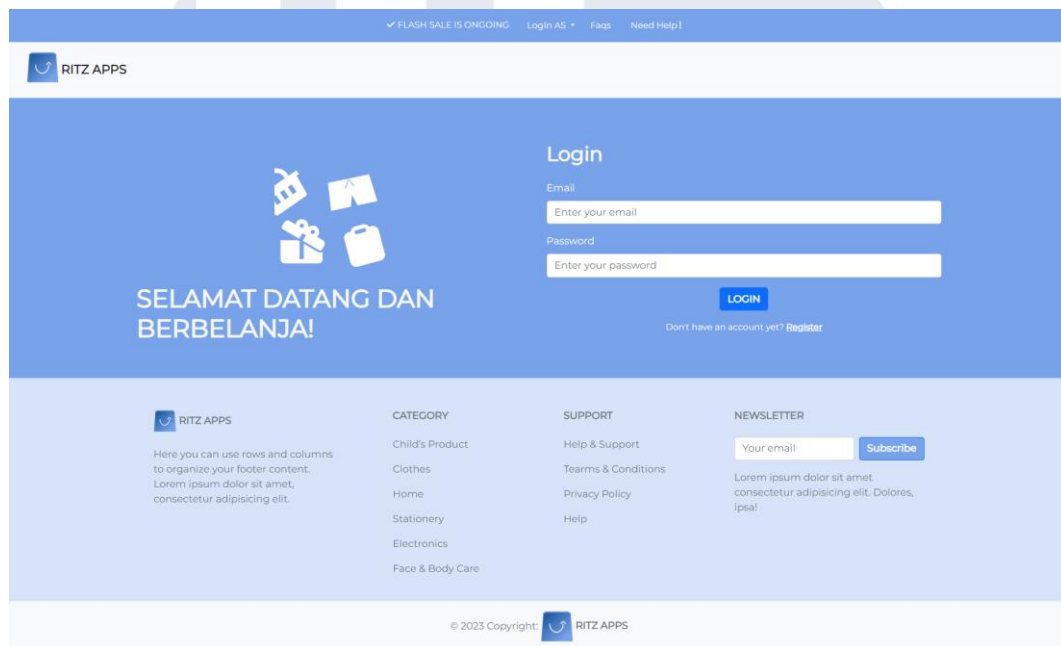
```

<!-- Login Form -->
<section id="LoginForm">
  <div class="container text-light">
    <div class="row align-items-center pt-5 pb-5">
      <div class="col-md-6 mt-5">
        
        <p class="h1 fw-bolder">SELAMAT DATANG DAN BERBELANJA!</p>
      </div>
      <div class="col-md-6">
        <div class="row">
          <p class="h2 fw-bold mt-3">Login</p>
        </div>
        <div class="row mt-3">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Email</label>
            <input type="email" class="form-control" id="email" placeholder="Enter your email">
          </div>
        </div>
        <div class="row">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Password</label>
            <input type="password" class="form-control" id="password" placeholder="Enter your password">
          </div>
        </div>
        <div class="row">
          <div class="col text-center">
            <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#modalCom
  
```

Gambar 3. 10 Coding front end login user

Pada gambar 3.10 di atas merupakan coding untuk halaman *login user*. Terdapat foto sebagai *background* halaman, disertai dengan alternatif yang bertuliskan “*Backgrund register*” fungsinya adalah ketika website sedang mengalami gangguan tulisan tersebut akan muncul sebagai alternatif dari gambar. Terdapat label yang bertuliskan “*Selamat datang dan berbelanja!*”, agar semakin terlihat tulisan tersebut ditambahkan dengan gaya *bold*. Terdapat *input box* yang digunakan untuk memasukan *email* dan *password*, di dalam *input box* tersebut terdapat *place holder* yang fungsinya untuk memberikan tulisan jika *input box* tersebut belum di isi. Pada halaman *login inputbox* email terdapat placeholder yang bertuliskan perintah untuk memasukan *email*, begitu pula dengan *input box*

password namun yang menjadi pembeda hanya tulisan untuk memasukan password. Pada bagian bawah terdapat button “LOGIN” yang berfungsi sebagai perintah *submit* apabila user telah memasukan *email dan password*. Berikut hasil *coding front end* tampilan login user.



Gambar 3. 11 Hasil Coding front end login uer

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

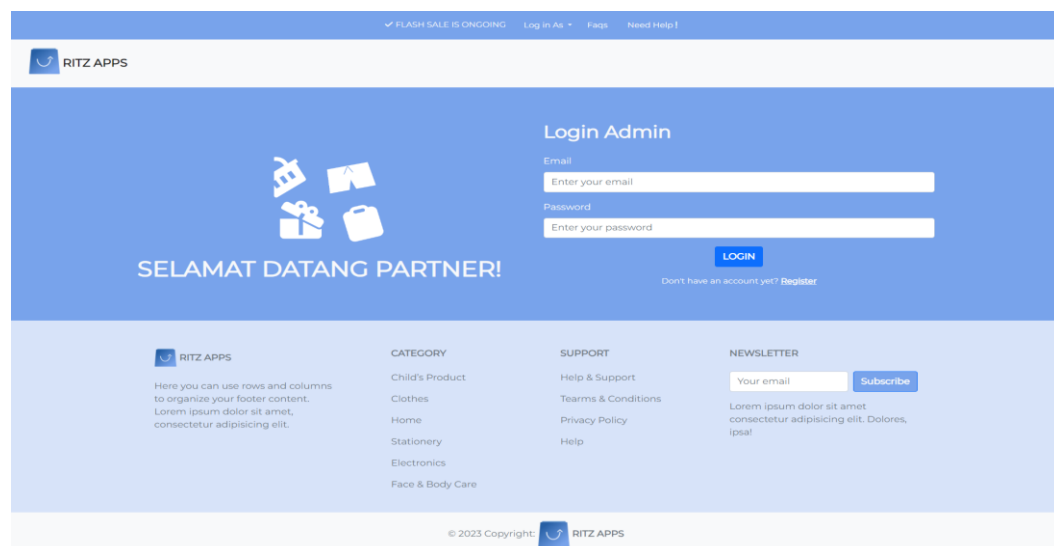
```

<!-- Login Form -->
<section id="LoginForm">
  <div class="container text-light">
    <div class="row align-items-center pt-5 pb-5">
      <div class="col-md-6 mt-5">
        SELAMAT DATANG PARTNER!</p>
      </div>
      <div class="col-md-6">
        <div class="row">
          <p class="h2 fw-bold mt-3">Login Admin</p>
        </div>
        <div class="row mt-3">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Email</label>
            <input type="email" class="form-control" id="email" placeholder="Enter your email">
          </div>
        </div>
        <div class="row">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Password</label>
            <input type="password" class="form-control" id="password" placeholder="Enter your password">
          </div>
        </div>
        <div class="row">
          <div class="col text-center">
            <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#modalCom

```

Gambar 3. 12 Coding front end login admin

Pada gambar 3.12 diatas merupakan coding front end untuk tampilan *login admin*. Coding tidak banyak memiliki perbedaan dengan tampilan *login user* yang berbeda. Untuk membedakan terdapat label bertuliskan “login Admin”, serta pada label ucapan selamat datang menjadi “Selamat datang partner”. Berikut hasil coding *front end login admin*.



Gambar 3. 13 Hasil coding front end login admin

```

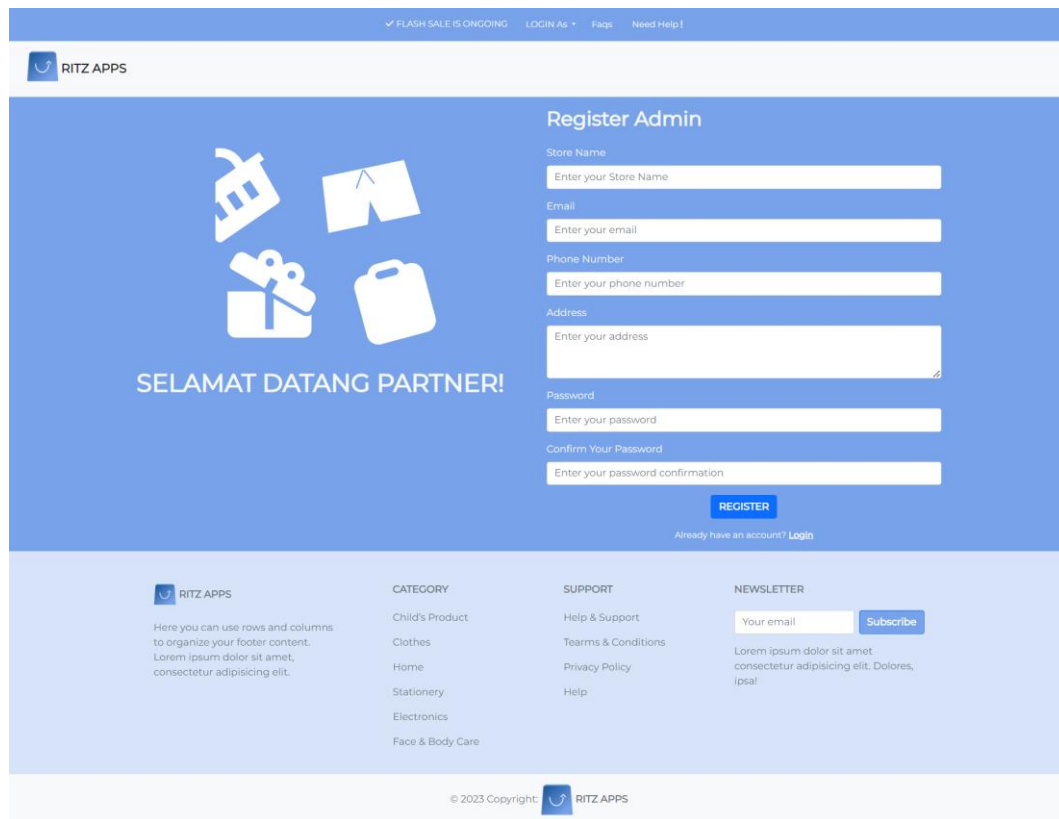
<!-- Register Form -->
<section id="RegisterForm">
  <div class="container text-light">
    <div class="row">
      <div class="col-md-6 mt-5">
        SELAMAT DATANG PARTNER!</p>
      </div>
      <div class="col-md-6">
        <div class="row">
          <p class="h2 fw-bold mt-3">Register Admin</p>
        </div>
        <div class="row mt-3">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Store Name</label>
            <input type="text" class="form-control" id="fullname" placeholder="Enter your Store Name">
          </div>
        </div>
        <div class="row">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Email</label>
            <input type="email" class="form-control" id="email" placeholder="Enter your email">
          </div>
        </div>
        <div class="row">
          <div class="mb-3">
            <label for="exampleFormControlInput1" class="form-label">Phone Number</label>
            <input type="number" class="form-control" id="phoneNumber" placeholder="Enter your phone number">
          </div>
        </div>
      </div>
    </div>
  </div>

```

Gambar 3.14 Coding front end register admin

Pada gambar 3.14 diatas merupakan coding front end untuk tampilan register admin. Terdapat gambar *background* tampilan website, di bawahnya terdapat label ucapan selamat datang bertuliskan “selamat datang partner”. Untuk membedakan dengan tampilan terdapat label “Register Admin”, dibawahnya ada enam *input box* pada coding diatas yaitu Store Name, Email, Phone Number, Address, Password, Confirm your password. Untuk *submit* inputan user terdapat button “Register”. Berikut hasil *coding front end register admin*.

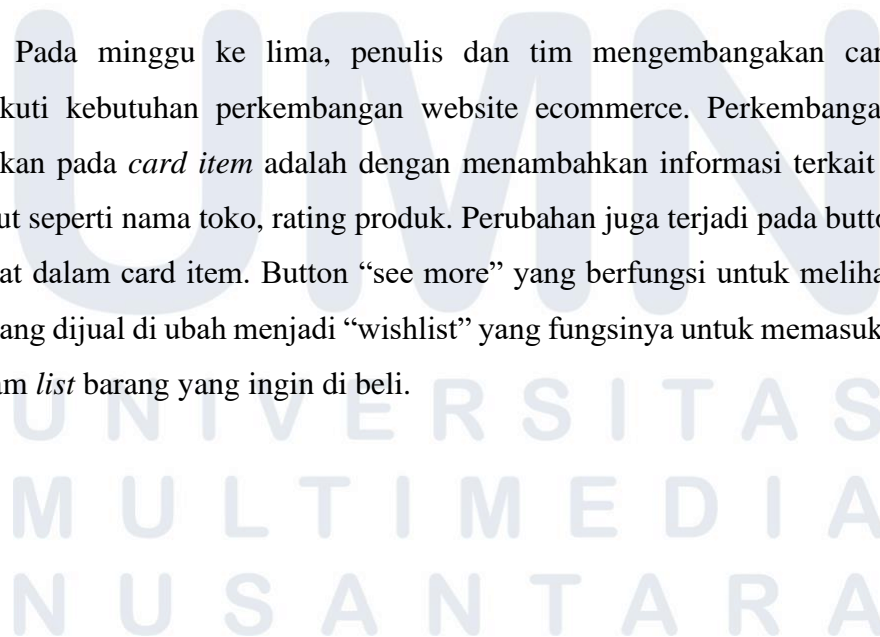
UNIVERSITAS
MULTIMEDIA
NUSANTARA

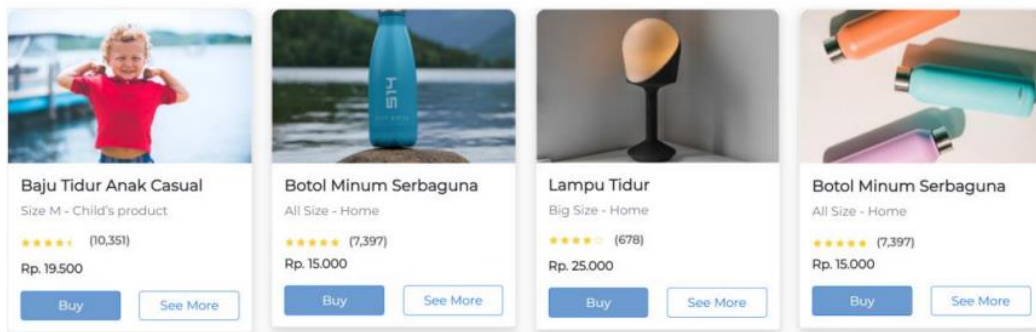


Gambar 3. 15 Tampilan front end register admin

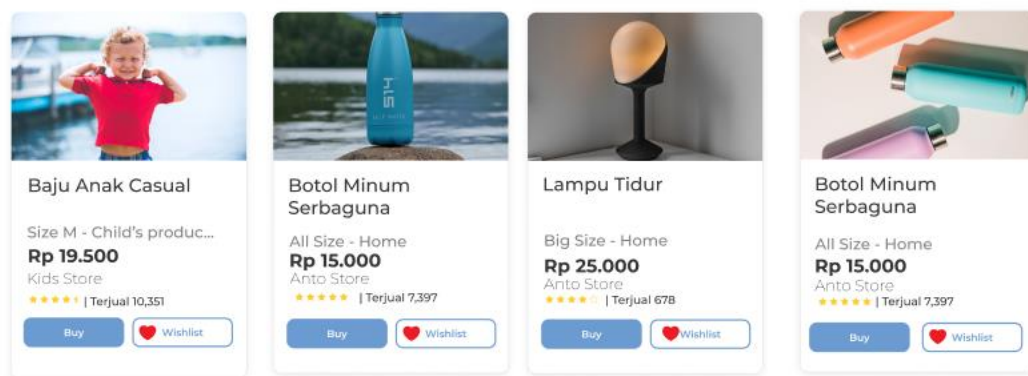
4. Update mockup UI/UX card item, dan Pembuatan mockup master event Pada 17 Juli 2023 – 21 Juli 2023

Pada minggu ke lima, penulis dan tim mengembangkan card item mengikuti kebutuhan perkembangan website ecommerce. Perkembangan yang dilakukan pada *card item* adalah dengan menambahkan informasi terkait produk tersebut seperti nama toko, rating produk. Perubahan juga terjadi pada button yang terdapat dalam card item. Button “see more” yang berfungsi untuk melihat detail item yang dijual di ubah menjadi “wishlist” yang fungsinya untuk memasukan item kedalam *list* barang yang ingin di beli.





Gambar 3. 16 Card item sebelum update



Gambar 3. 17 Mockup card item update

Pada gambar 3.16 di atas merupakan card item sebelum update. Pada card item sebelum update belum terdapat nama toko karena pada awalnya website hanya digunakan untuk satu toko saja sehingga seluruh item yang dijual oleh satu toko. Pada gambar 3.17 merupakan mockup card dengan update dengan pengembangan nama toko, selain penambahan nama toko terdapat perubahan susunan informasi item pada card. Font harga pada card di pertebal agar semakin mudah untuk melihat harga. Perubahan lain terjadi pada button “see more” yang di ubah menjadi wishlist. Setelah pembuatan mockup langkah selanjutnya adalah coding front end card item.

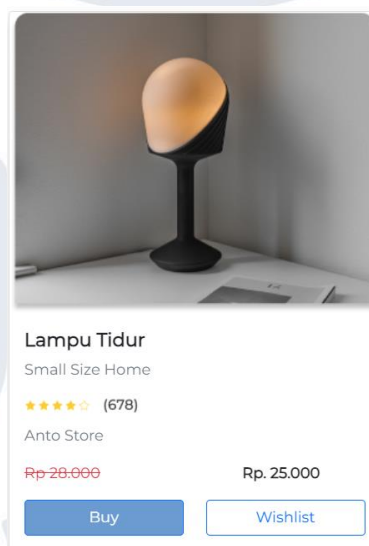
```

<div class="col-md">
  <div class="card w-100 shadow" style="width: 18rem">
    
    <div class="card-body">
      <h5 class="card-title fw-bold">Lampu Tidur</h5>
      <p class="text-muted card-text">Big Size - Home</p>
      
      <p class="fw-bold">Rp. 25.000</p>
      <div class="row">
        <div class="col-md">
          <button
            type="button"
            class="btn btn-outline-primary d-block w-100 text-white"
            style="background-color: #6b9bd0"
          >
            Buy
          </button>
        </div>
        <div class="col-md text-end">
          <button
            type="button"
            class="btn btn-outline-primary d-block w-100"
          >
            Wishlist
          </button>
        </div>
      </div>
    </div>
  </div>

```

Gambar 3. 18 Coding card item

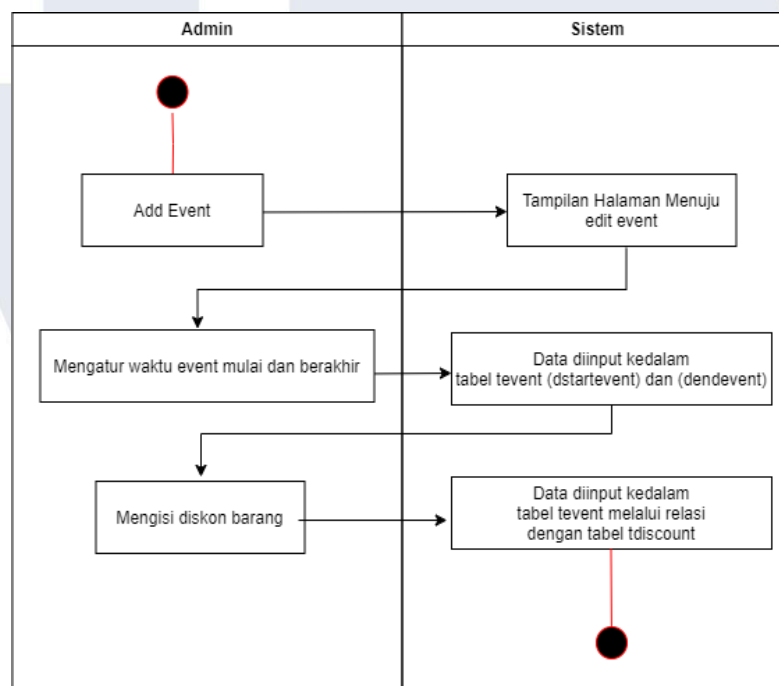
Pada gambar 3.18 merupakan coding untuk card item. Coding belum terhubung dengan database sehingga informasi produk menggunakan *hard code*. *Card item* yang sudah terhubung dengan database di kerjakan oleh tim *back end*.



Gambar 3. 19 Hasil front end card item

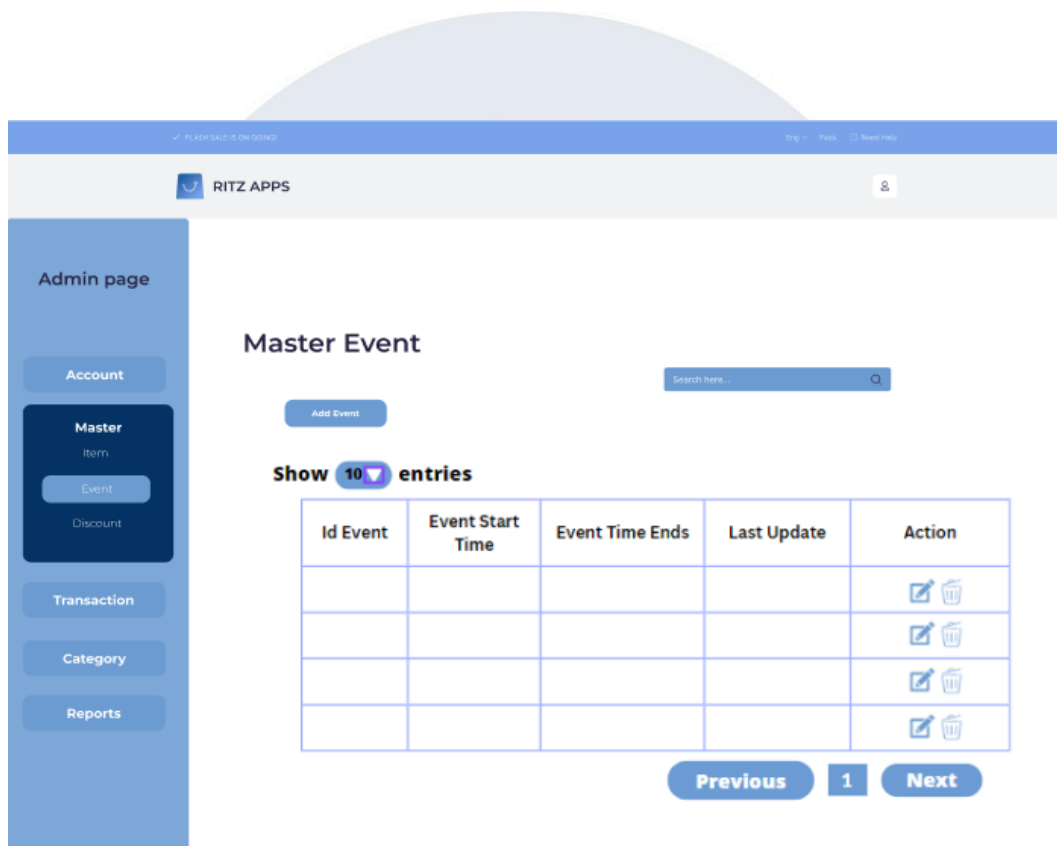
Pada gambar 3.19 diatas merupakan hasil coding front end card item yang di buat menggunakan HTML, CSS, dan bootstrap 5.

Pada fase ini penulis dan tim *system analyst* menambahkan fitur baru untuk admin yaitu master event. Master event adalah sebuah fitur untuk toko yang fungsinya untuk membuat *event* atau acara, dalam *event* tersebut nantinya akan terdapat diskon yang bisa dibuat oleh toko.



Gambar 3. 20 Activity diagram Master event

Pada gambar 3.20 di atas merupakan activity diagram master event. Langkah pertama, aktor menambahkan event kemudian sistem akan mengarahkan ke tampilan edit event, selanjutnya aktor mengatur waktu mulai dan berakhirnya event lalu sistem akan menyimpan ke dalam tabel “tevent”, kemudian aktor bisa menambahkan diskon barang dan sistem akan menyimpan ke dalam tabel “tdiscout”.



Gambar 3. 21 Mockup Master event

Pada gambar 3.21 merupakan mockup master event yang di buat menggunakan figma. Pada tampilan tersebut terdapat *side bar* untuk memilih tampilan, terdapat button “add event” untuk menambahkan event, *search bar* untuk mencari event yang sudah ditambahkan. Terdapat tabel untuk menyimpan data event, ada lima kolom yaitu idEvent, start time, end time, last update, dan action. Terdapat dua ikon untuk *update* dan *delete*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

5. Pembuatan front end master event, dan Pengerjaan bagian pemisahan hak akses *user, admin*, dan super admin Pada 24 Juli 2023 – 29 Juli 2023

Pada minggu ke lima, penulis membuat tampilan front end master event. Tampilan di buat sesuai dengan rancangan mockup yang telah di buat. Tampilan front end yang dibuat merupakan sebuah template yang belum terhubung dengan *database*. Tampilan yang dibuat pada *front end* bisa mengalami sedikit perubahan menyesuaikan tingkat kesulitan.

```
<section id="accountManagement">
  <div class="container-fluid overflow-hidden border-top">
    <div class="row vh-auto overflow-auto" style="background-color: #white;">
      <div class="col-12 col-sm-3 col-xl-2 px-sm-2 px-0 d-flex sticky-top" style="background-color: #d7e3f9;">
        <div class="d-flex flex-sm-column flex-row flex-grow-1 align-items-center align-items-sm-start px-3 pt-3">
          <a href="/" class="d-flex pb-sm-3 mb-md-0 me-md-auto text-dark text-decoration-none fw-bolder text h3">
            <ul class="nav nav-pills flex-sm-column flex-row flex-nowrap flex-shrink-1 flex-sm-grow-0 flex-grow-1 m">
              <li>
                <a href="#" class="nav-link px-sm-0 px-2">
                  <i class="fs-5 bi-table"></i><span class="ms-1 d-none d-sm-inline">Account Management</span></a>
              </li>
              <li>
                <a href="#" class="nav-link active px-sm-0 px-2">
                  <i class="fs-5 bi-table"></i><span class="ms-1 d-none d-sm-inline">Master Event</span></a>
              </li>
              <li>
                <a href="#" class="nav-link px-2">
                  <i class="fs-5 bi-table"></i><span class="ms-1 d-none d-sm-inline">Master Discount</span></a>
              </li>
              <li>
                <a href="#" class="nav-link px-sm-0 px-2">
                  <i class="fs-5 bi-table"></i><span class="ms-1 d-none d-sm-inline">Member Request</span> </a>
              </li>
              <li>
                <a href="#" class="nav-link px-sm-0 px-2">
                  <i class="fs-5 bi-table"></i><span class="ms-1 d-none d-sm-inline">Item Header</span></a>
              </li>
              <li>
                <a href="#" class="nav-link px-sm-0 px-2">
                  <i class="fs-5 bi-table"></i><span class="ms-1 d-none d-sm-inline">Item Detail</span></a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </section>
```

Gambar 3. 22 Coding Front end sidebar

Pada gambar 3.22 diatas merupakan coding untuk *sidebar*. Fungsi dari sidebar adalah untuk berpindah dari satu halaman ke halaman yang lain. *Coding sidebar* ini akan digunakan pada seluruh tampilan *front end* admin.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

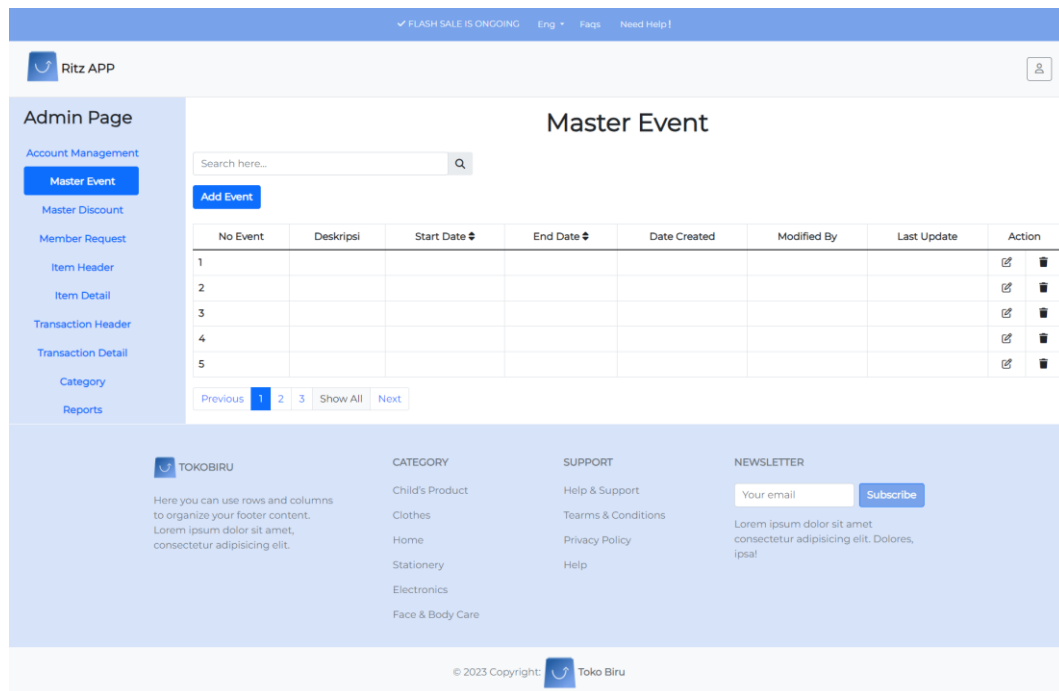
<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Master Event</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." aria-label="Search">
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
      <button type="button" class="btn btn-primary mt-3"><b>Add Event</b></button>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">No Event</th>
            <th scope="col">Deskripsi</th>
            <th scope="col">Start Date<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col">End Date<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col">Date Created</th>
            <th scope="col">Modified By</th>
            <th scope="col">Last Update</th>
            <th scope="col" colspan="2">Action</th>
          </tr>
        </thead>
      </table>
    </div>
  </main>
</div>

```

Gambar 3. 23 Coding front end master event

Pada gambar 3.23 di atas merupakan coding front end master event. Untuk menambahkan data event terdapat button dengan label “add event”. Kemudian, terdapat struktur tabel ditambah dengan *class table border* agar terdapat garis pada seluruh isi tabel, selanjutnya terdapat tabel row dengan tanda `<tr>`, tujuannya untuk membuat bariss pada kolom, didalamnya terdapat tabel head dengan tanda `<th>` tujuannya untuk membuat judul pada setiap kolom. Jumlah tabel head yang terdapat pada tabel adalah 8, salah satu tabel head digunakan untuk 2 action button sehingga kolom dibagi menjadi 2 bagian menggunakan `colspan`. Terdapat class text center agar setiap header kolom berada pada posisi tengah. Untuk menambah fitur pada tampilan terdapat *search bar* untuk mempermudah mencari data, juga terdapat page navigation yang disertai dengan *previous* dan *next* button fungsinya untuk menunjukkan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan master event.





Gambar 3. 24 front end tampilan master event

Penulis dan tim system analyst melakukan diskusi terkait dengan hak akses yang dapat di berikan kepada setiap *role*. Akun admin toko tidak selalu di operasikan oleh pemilik toko namun terkadang ada karyawan yang mengoperasikannya, terdapat kekhawatiran penyalahgunaan akun *admin* toko untuk melakukan perbelanjaan menggunakan saldo hasil penjualan dari akun tersebut. Setelah melakukan diskusi diputuskan bahwa akun admin tidak dapat mengakses halaman belanja website. *Role* user tidak mendapatkan hak untuk mengakses seluruh halaman admin. Pada fase ini tim *system analyst* memutuskan untuk menambahkan *role* super admin, dimana role super admin merupakan pemilik dari website ecommerce. Fungsi dari *role* super admin adalah memantau dan mengontrol website *ecommerce*, namun terdapat halaman yang tidak bisa di akses untuk menjaga kerahasiaan informasi akun pengguna. Terdapat 3 *role* dalam website *ecommerce* dengan akses pada masing masing role sebagai berikut:

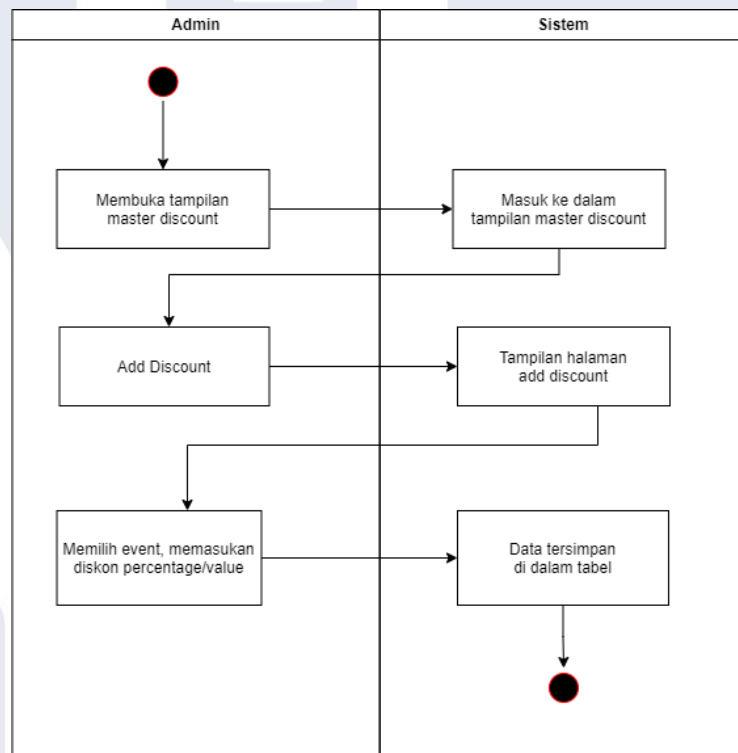
Menu	Submenu	Super Aadmin	Admin	User
Account	Account Managemet	Y	Y	
	Toko	Y	Y	
	Member Management	Y	Y	
Profile	Profile	Y	Y	Y
	Member Request	Y		Y
Setting	Global Paramenter	Y		
Master	Item Management	Y	Y	
	Item Detail	Y	Y	
	Item Catergory	Y	Y	
	Event	Y	Y	
	Discount	Y	Y	
	Holiday		Y	
	Operational		Y	
	Retur	Y	Y	
	Sales	Y	Y	
Transaction	Management	Y	Y	
	Detail	Y	Y	
	Event	Y		Y
	Report	Y	Y	
	Cart	Y		Y
	Checkout	Y		Y
	Transaction History	Y	Y	Y
	Detail Order	Y		Y
Payterm		Y		Y
Saldo		Y		Y
Poin		Y		Y
Retur		Y		Y
Return History				Y
Category		Y	Y	
Report		Y	Y	

Gambar 3. 25 Hak akses setiap role

UNIVERSITAS
MULTIMEDIA
NUSANTARA

6. Pembuatan activity diagram master diskon, Mockup master diskon, dan Front end master diskon Pada tanggal 31 Juli 2023 – 5 Agustus 2023

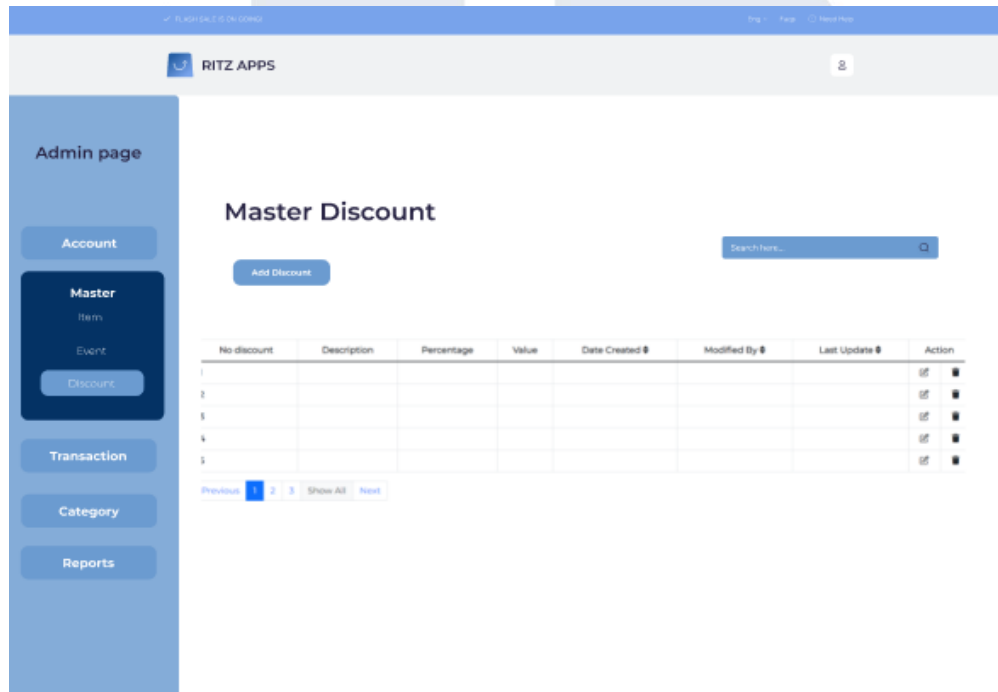
Pada fase ini, penulis dan tim menambahkan fitur baru bernama master discount. Master diskon adalah sebuah fitur pada role admin yang bisa memberikan potongan harga barang. *Discount* yang diberikan dalam dua macam yaitu berdasarkan presentasi dan *value* harga.



Gambar 3. 26 Activity diagram master discount

Pada gambar di 3.26 atas merupakan activity diagram untuk menambah diskon. Langkah pertama aktor membuka tampilan master diskon melalui *side navigation bar*, selanjutnya sistem akan mengarahkan *admin* kedalam tampilan master discount, kemudian aktor akan menekan *button add discount* untuk memasukan *discount* barang, lalu sistem akan mengarahkan ke tampilan *add discount*, selanjutnya aktor akan memasukan *discount* barang dengan dua pilihan

antara *discount* dengan persentase atau *value*. Sistem akan menyimpan seluruh informasi *discount* ke dalam *database* dan menampilkannya pada tabel discount.



Gambar 3. 27 Mockup master discount

Pada gambar 3.27 diatas merupakan mockup tampilan master *discount* yang dibuat menggunakan figma. Terdapat *side navigation* bar untuk memilih dan berpindah tampilan. Terdapat tabel yang terdiri dari delapan kolom yang berisi no. discount, description, percentage, value, date created, modified by, last update, dan ikon. Pada bagian atas tabel terdapat button add discount dan juga search bar. Terdapat page navigation untuk berganti halaman tabel.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

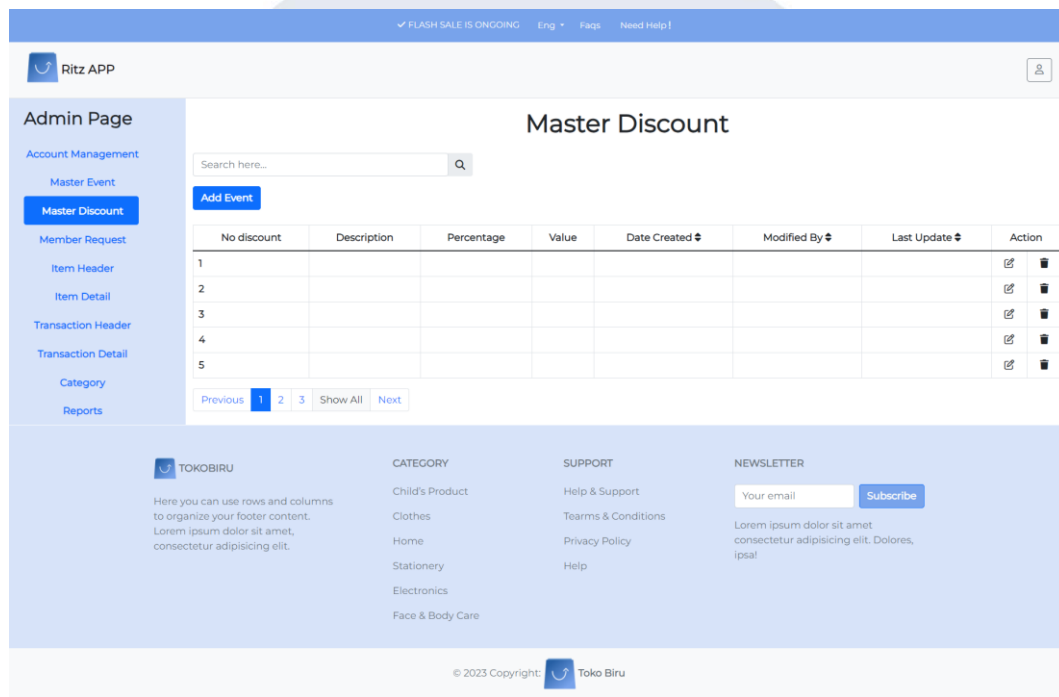
</div>
<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Master Discount</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." aria-label="Search" />
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
      <button type="button" class="btn btn-primary mt-3"><b>Add Event</b></button>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered" width="1000px">
        <thead>
          <tr class="text-center">
            <th scope="col">No discount</th>
            <th scope="col">Description</th>
            <th scope="col">Percentage</th>
            <th scope="col">Value</th>
            <th scope="col">Date Created<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col">Modified By<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col">Last Update<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col" colspan="2">Action</th>
          </tr>
        </thead>
      </table>
    </div>
  </main>
</div>

```

Gambar 3. 28 Coding front end master discount

Pada gambar 3.28 diatas merupakan *front end coding* tampilan master discount. Terdapat label master discount dengan *class bold*. Untuk menambahkan data *discount* terdapat *button* dengan label “add event”. Selanjutnya untuk membuat tabel menggunakan struktur tabel ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat baris pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 8, salah satu tabel head digunakan untuk 2 *action button update, delete* sehingga kolom dibagi menjadi 2 bagian menggunakan *colspan*. Pada tabel *row* terdapat *class text center* agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada bagian bawah tabel terdapat *page navigation* yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukkan

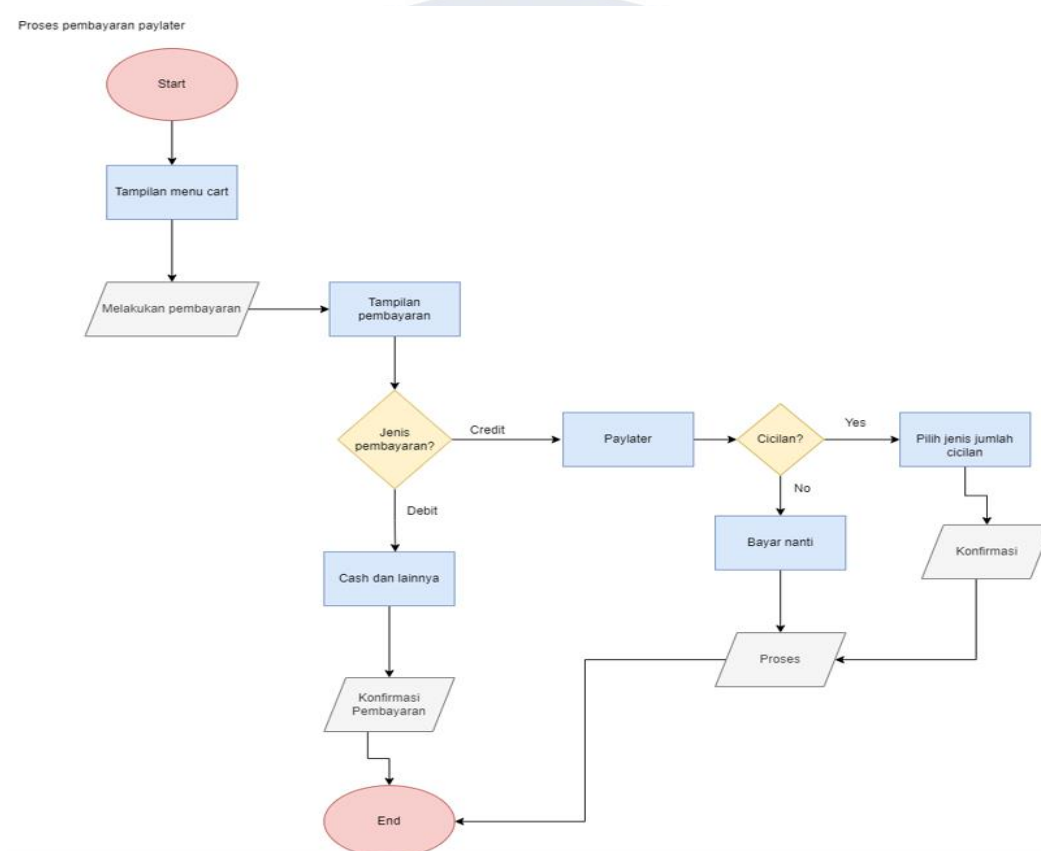
halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan master event:



Gambar 3. 29 Hasil front end master discount

7. Pembuatan flowchart, dan activity diagram fitur paylater pada tanggal 7 Agustus 2023 - 12 Agustus 2023

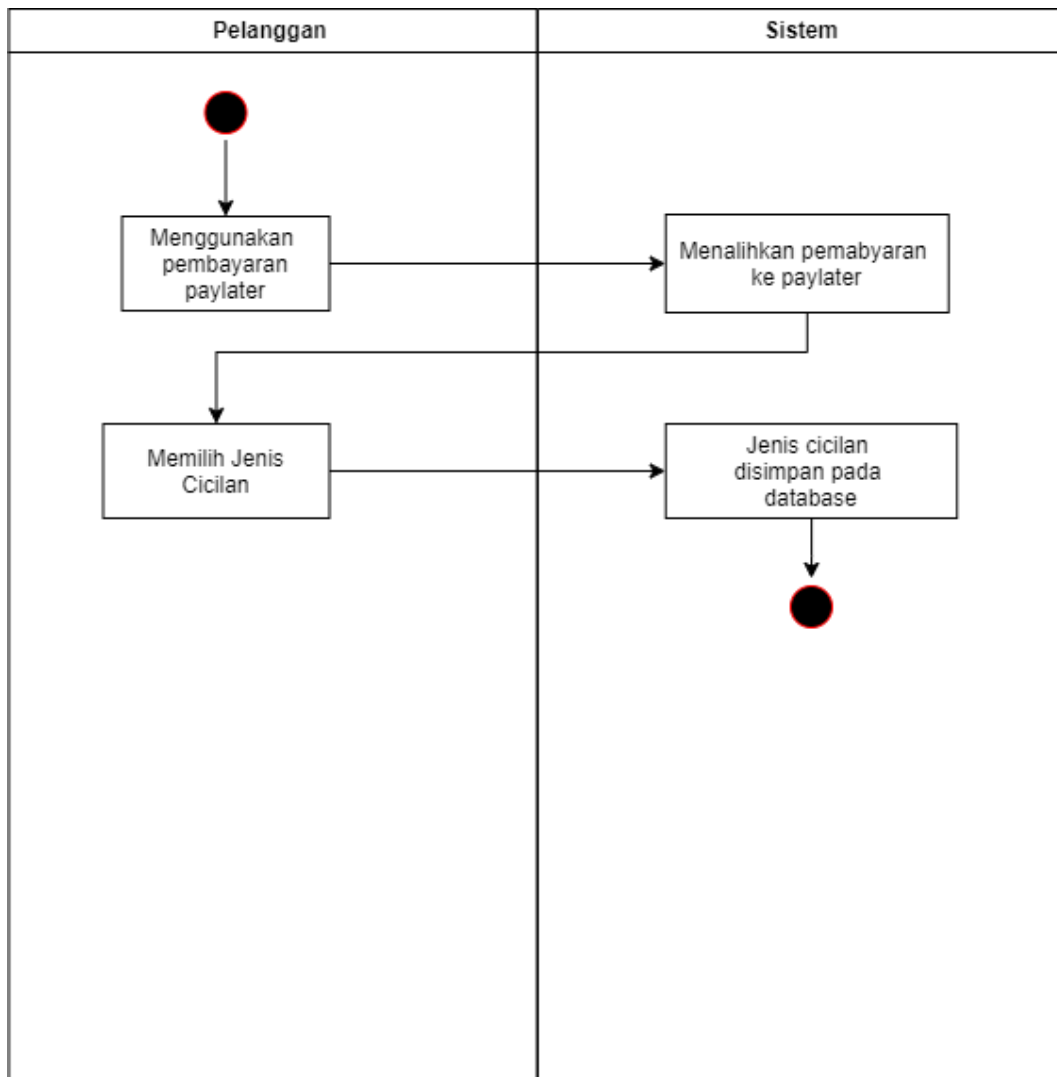
Pada minggu ke tujuh, penulis dan tim melakukan pembahasan untuk membuat fitur pay later. Pay later adalah fitur yang dapat digunakan oleh pengguna dalam melakukan transaksi pembayaran dimana pengguna tidak harus membayar pada saat proses check *out* namun bisa membayar pada limit waktu yang sudah ditentukan, serta dengan proses pelunasan yang bisa dicicil. Untuk dapat menggunakan fitur pay later pengguna harus mendaftarkan akunya menjadi member terlebih dahulu, akun yang dapat menjadi member harus mendapat persetujuan langsung dari admin. *Limit* pembayaran yang dapat digunakan akan disesuaikan dengan poin yang terdapat pada akun. Poin akun didapatkan dari setiap transaksi pembelian yang telah berhasil.



Gambar 3. 30 Flowchart fitur paylater

Pada gambar 3.30 diatas merupakan alur fitur *paylater*. Alur dimulai dari tampilan menu cart, selanjutnya melakukan pembayaran lalu diarahkan ke tampilan pembayaran. Pembayaran dilakukan dengan *cash* atau *paylater*, apabila menggunakan *cash* akan langsung melakukan konfirmasi pembayaran, jika menggunakan pay later pengguna harus memilih tipe pay later antara pembayaran tagihan secara full atau dengan cicilan. selanjutnya transaksi di proses dan selesai.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

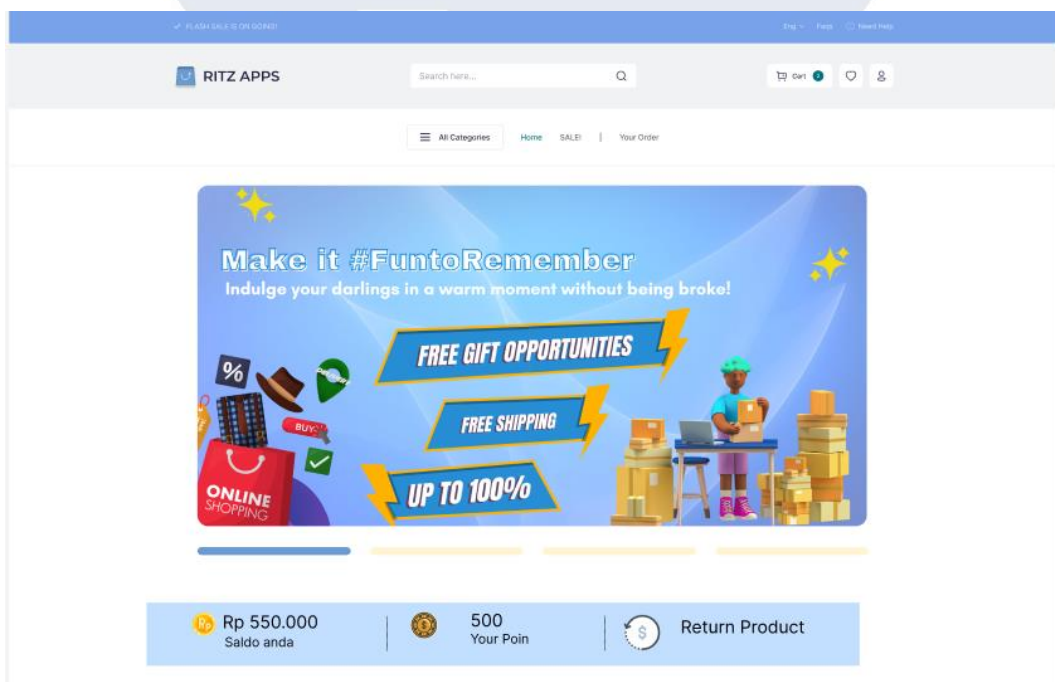


Gambar 3. 31 Activity diagram pay later.

Pada gambar 3.31 diatas merupakan *activity* diagram pelanggan yang menggunakan fitur pay later. Langkah pertama pelanggan melakukan transaksi pembelian dengan menggunakan fitur pembayaran *paylater*, selanjutnya sistem akan mengarahkan ke dalam tampilan pay later, kemudian pelanggan akan memilih jenis cicilan yang akan digunakan, selanjutnya sistem akan menyimpan jenis cicilan yang telah di pilih pelanggan ke dalam database.

8. Pengerjaan Mockup dan Front end informasi poin, Master Operatioanal Hour, Master Holiday, Update tampilan detail item pada tanggal 14 Agustus 2023 – 26 Agustus 2023

Pada tahap ini, penulis mendapatkan tugas untuk menempatkan posisi informasi poin. Fitur poin merupakan fitur lanjutan dari fitur pay later untuk menentukan limit pembayaran yang dapat digunakan. Fungsi dari informasi poin yang dibuat untuk menampilkan jumlah poin yang telah dimiliki pada akun tersebut. Langkah yang dilakukan adalah membuat mockup penempatan posisi fitur poin. Mockup yang telah di buat akan di masukan ke dalam dokumen spesifikasi program (SP) yang nantinya akan digunakan sebagai acuan pembuatan tampilan pada tampilan *front end*.



Gambar 3. 32 Mockup penempatan informasi poin

Pada gambar 3.32 diatas merupakan mockup penempatan informasi poin. Pada mockup diatas penempatan informasi poin berada pada halaman home page, terdapat ikon poin untuk meningkatkan estetika dan *awareness* pengguna. Pemilihan penempatan pada halaman home page agar pengguna dapat dengan

mudah mengetahui jumlah poin yang dimiliki. Pada saat pembuatan mockup penulis merasa masih terdapat banyak space kosong jika hanya menempatkan informasi poin sehingga penulis menempatkan fitur dan informasi lain yang masih dalam tahap wacana. Informasi saldo ditempatkan pada tempat yang sama dengan info saldo, dan fitur retur produk. Mockup yang dibuat telah di setujui oleh tim *system analyst* dan *supervisor*, selanjutnya pembuatan tampilan dari sisi *front end*.

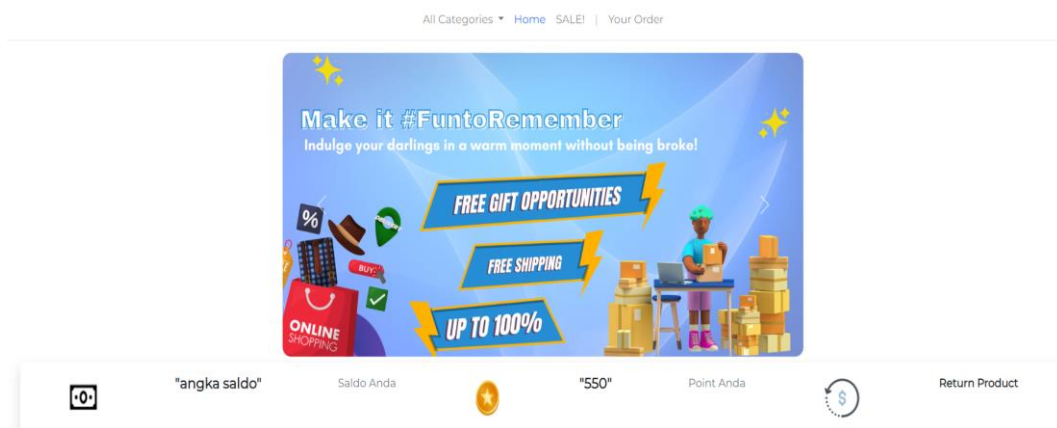
```
<div class="row shadow bg-body rounded m-2">
  <div class="col-md p-3 d-flex justify-content-around">
    
    <h5 class="card-title fw-bold">"angka saldo"</h5>
    <p class="text-muted card-text">Saldo Anda</p>

    
    <h5 class="card-title fw-bold">"550"</h5>
    <p class="text-muted card-text">Point Anda</p>
    
    <p class="card-title fw-bold">Return Product</p>
  </div>
</div>
```

Gambar 3. 33 Coding front end informasi saldo

Pada gambar 3.33 di atas merupakan coding front end untuk informasi saldo. Penempatan coding berada pada tampilan home page sehingga gambar diatas merupakan potongan codingan pada front end home page. Terdapat class "row

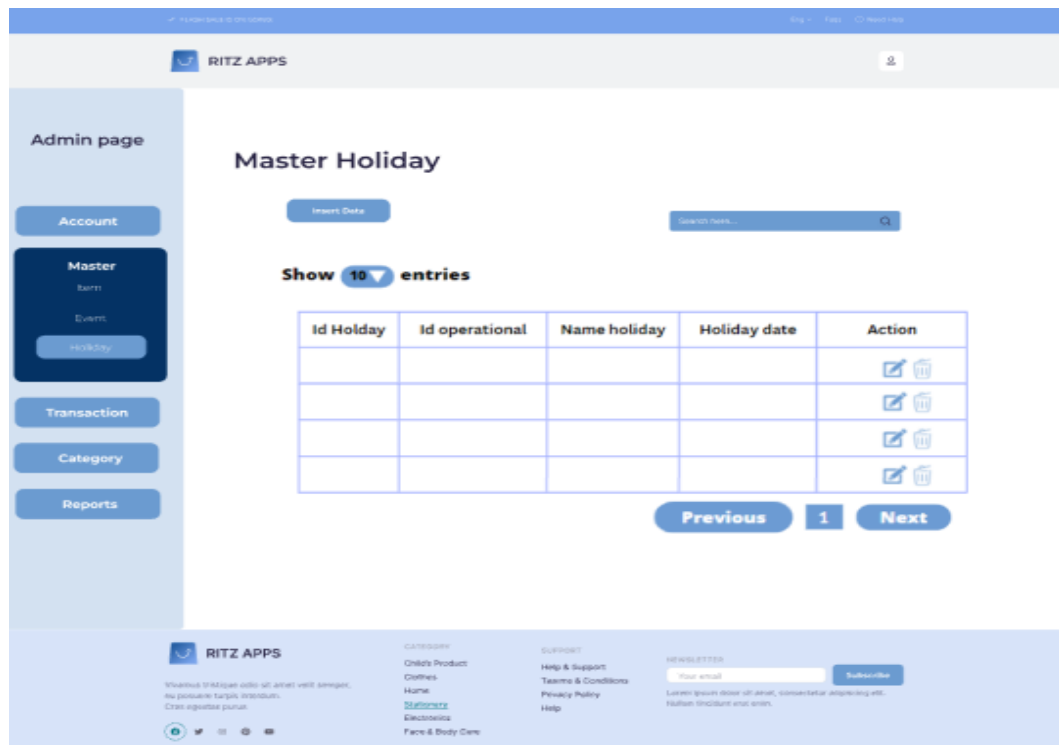
shadow bg-body rounded m-2" untuk membuat bariss dengan background. Di dalam class bariss tersebut terdapat sub class *"col-md p-3 d-flex justify-content-around"* untuk membagi bariss menjadi tiga bagian, pada setiap bagian tersebut terdapat gambar untuk ikon, informasi. Pada tampilan front end informasi yang tercantum belum terhubung dengan database yang nantinya akan di kerjakan oleh tim *back end*, sehingga informasi yang tercantum dibuat dalam bentuk *hard code*. Berikut adalah hasil tampilan coding informasi saldo:



Gambar 3. 34 hasil coding front end penempatan info saldo.

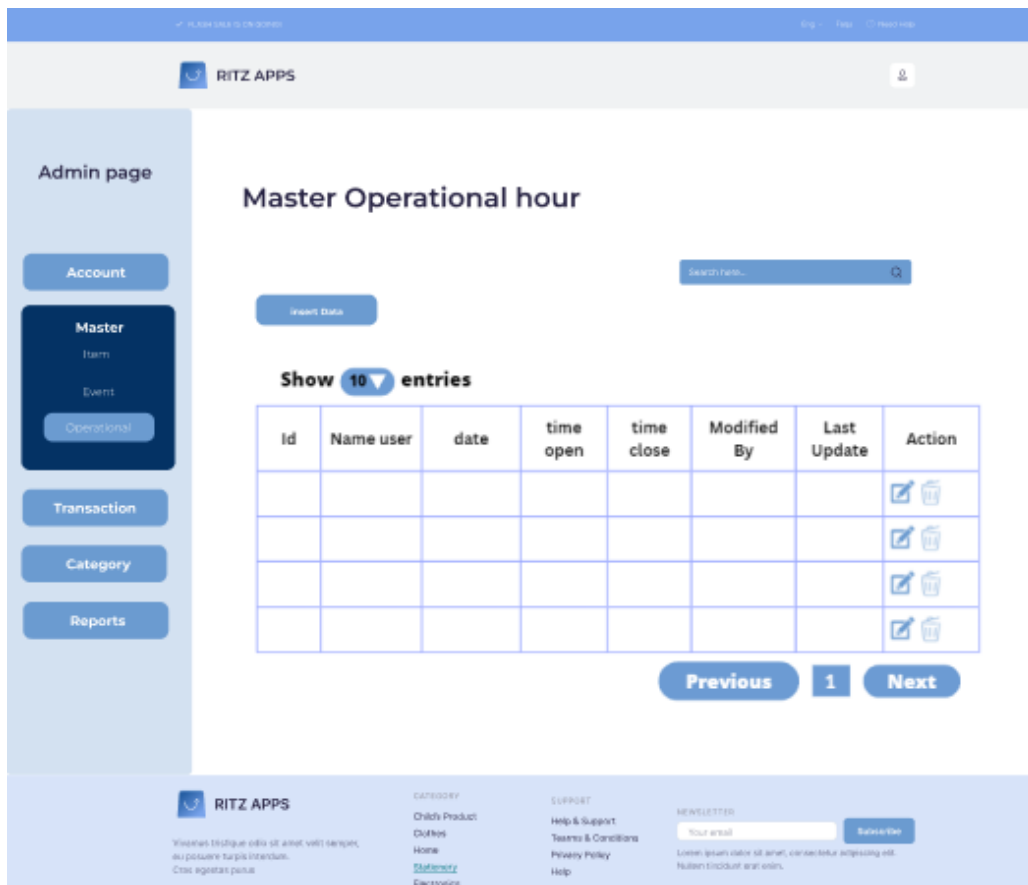
Setelah membuat informasi saldo, penulis mendapatkan tugas membuat tampilan mockup dengan figma dan front end untuk fitur master holiday dan master operational hour. Master holiday adalah fitur untuk toko dimana fungsinya untuk mengatur libur toko dan tidak ada yang bisa membeli produk dari toko tersebut apabila termasuk ke dalam hari libur. Master operational hour adalah fitur untuk toko yang memiliki fungsi untuk mengatur waktu operasional toko (jam buka hingga tutup) sehingga dapat digunakan sebagai informasi untuk user kapan waktu operasional toko. Berikut adalah mockup tampilan figma master holiday dan master operational hour:

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 35 Mockup master holiday

Pada gambar 3.35 merupakan mockup master holiday yang di buat menggunakan figma. Pada tampilan tersebut terdapat *side bar* untuk memilih tampilan, terdapat *button* “insert data” untuk menambahkan hari libur. kemudian terdapat search bar untuk mencari data yang sudah ditambahkan. Terdapat tabel untuk menyimpan data holiday, ada lima kolom yaitu id Holiday, id operational, name holiday, holiday date, dan action. Terdapat dua ikon untuk *update* dan *delete*. Terdapat page navigation untuk berpindah halaman tabel.



Gambar 3. 36 Mockup master operational hour

Pada gambar 3.36 merupakan mockup master operational hour yang di buat menggunakan figma. Tampilan tidak memiliki banyak perbedaan dari master holiday. Pada tampilan tersebut terdapat *side bar* untuk memilih tampilan, terdapat button “insert data” untuk menambahkan waktu operasional toko. kemudian terdapat search bar untuk mencari data yang sudah ditambahkan. Terdapat tabel untuk menyimpan data operational hour, ada delapan kolom yaitu id, name user, date, time open, time close, modified by, last update, dan action. Terdapat dua ikon untuk *update* dan delete. Terdapat page navigation untuk berpindah halaman tabel.

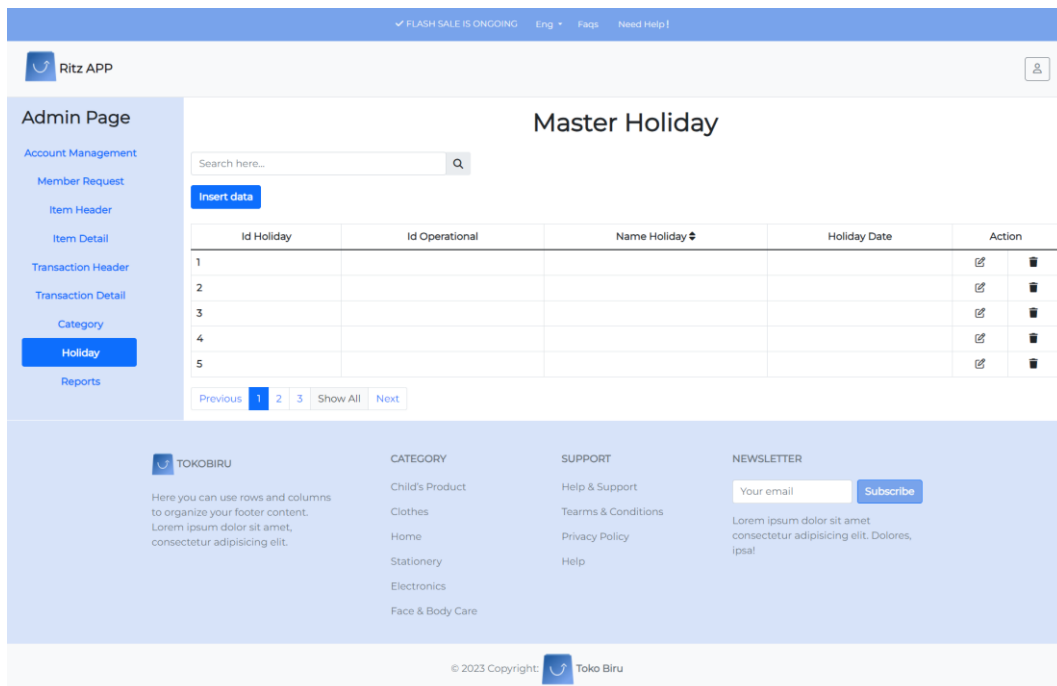
```

<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Master Holiday</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." aria-label="Search">
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
      <button type="button" class="btn btn-primary mt-3"><b>Insert data</b></button>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">Id Holiday</th>
            <th scope="col">Id Operational</th>
            <th scope="col">Name Holiday<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col">Holiday Date</th>
            <th scope="col" colspan="2">Action</th>
          </tr>
        </thead>

```

Gambar 3. 37 Coding front end master holiday

Pada gambar 3.36 diatas merupakan *front end coding* tampilan master holiday. Terdapat label master discount dengan *class bold*. Untuk menambahkan data *discount* terdapat *button* dengan label “insert data”. Selanjutnya untuk membuat tabel menggunakan struktur tabel ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat bariss pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 5, salah satu tabel head digunakan untuk 2 *action button update, delete* sehingga kolom dibagi menjadi 2 bagian menggunakan *colspan*. Pada tabel row `<tr>` terdapat class text center agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada bagian bawah tabel terdapat page navigation yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan master holiday.



Gambar 3. 38 Hasil coding front end tampilan master holiday

Setelah membuat coding front end untuk tampilan master holiday selanjutnya penulis membuat front end coding untuk master operational hour. Codingan tidak banyak berbeda dari master holiday, yang membedakan hanya pada table head <th>. Berikut adalah coding tampilan master operational:

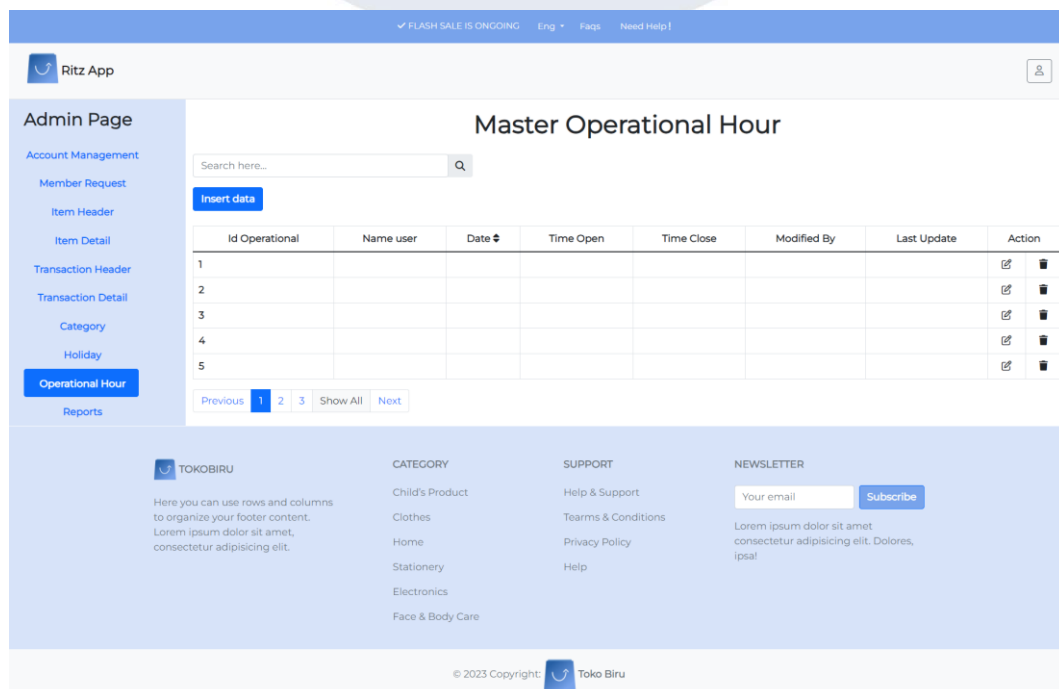
```

<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Master Operational Hour</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." aria-label="Search">
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
      <button type="button" class="btn btn-primary mt-3"><b>Insert data</b></button>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">Id Operational</th>
            <th scope="col">Name user</th>
            <th scope="col">Date<i class="fa-solid fa-sort ms-1"></i></th>
            <th scope="col">Time Open</th>
            <th scope="col">Time Close</th>
            <th scope="col">Modified By</th>
            <th scope="col">Last Update</th>
            <th scope="col" colspan="2">Action</th>
          </tr>

```

Gambar 3. 39 Coding front end master operational hour

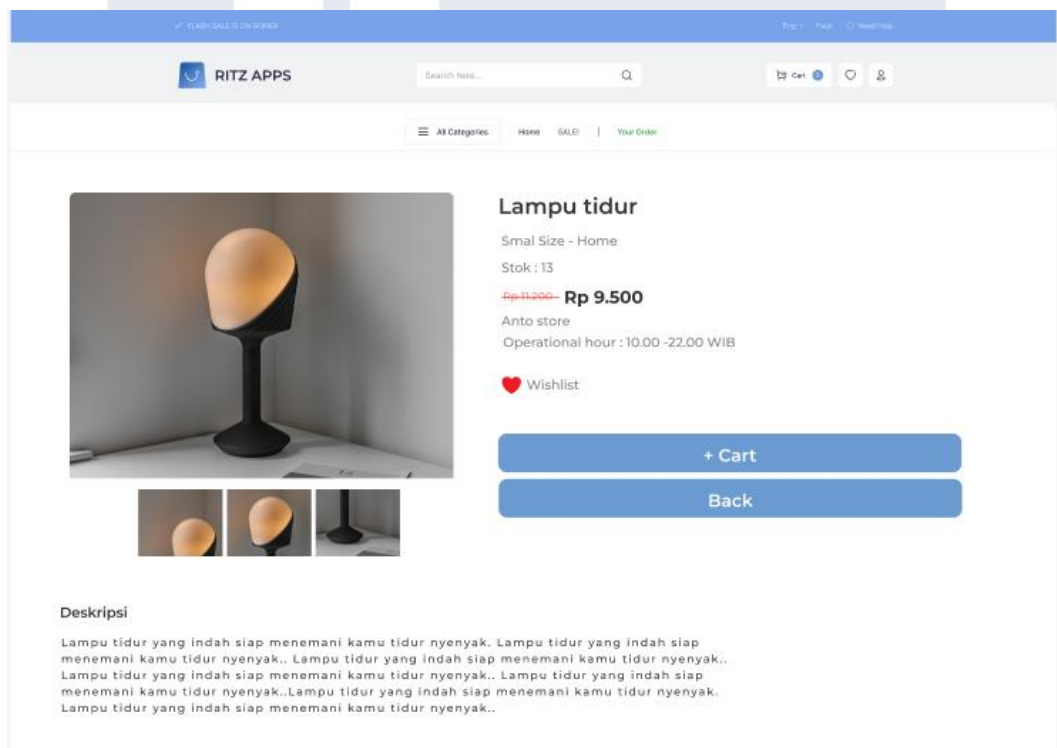
Pada gambar 3.39 merupakan *front end coding* tampilan master operational hour. Terdapat label master discount dengan *class bold*. Untuk menambahkan data *discount* terdapat *button* dengan label “insert data”. Selanjutnya untuk membuat tabel menggunakan struktur tabel ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat baris pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 8, salah satu tabel head digunakan untuk 2 *action button update, delete* sehingga kolom dibagi menjadi 2 bagian menggunakan *colspan*. Pada tabel *row <tr>* terdapat *class text center* agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada bagian bawah tabel terdapat *page navigation* yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan master operational hour.



Gambar 3. 40 Hasil coding front end tampilan master operational hour.

9. Pembuatan mockup, front end penempatan informasi operational hour, dan Pengerjaan Front end Master Sales pada 28 Agustus 2023 – 2 September 2023

Pada tahap ini, penulis telah mengerjakan fitur untuk master operational hour untuk sisi admin. Fitur operational hour juga berfungsi sebagai informasi untuk user kapan waktu operational dari toko. Penulis memberikan update penempatan posisi informasi operational hour toko. Berikut adalah *mockup* tampilan penempatan informasi operational hour toko yang di buat menggunakan figma:



Gambar 3. 41 Mockup penempatan informasi operational hour

Pada gambar 3.41 di atas merupakan mockup tampilan item detail. Informasi operational hour toko di tempatkan pada halaman item detail, penempatan di halaman ini agar pengguna semakin *aware* terkait jam operational toko. Selanjutnya penulis akan melakukan update *coding* untuk informasi operational hour. Berikut coding *front end*:

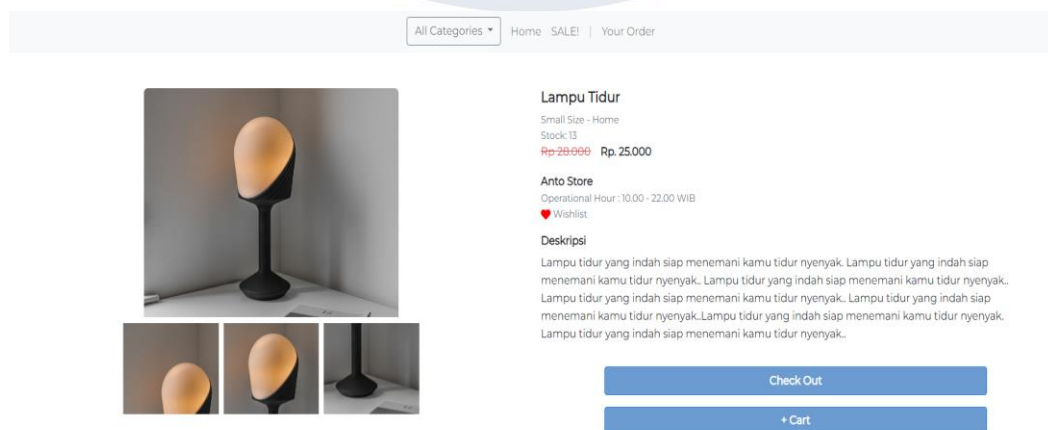
OW

```
<div class="col-md mt-5 text-start">
  <p class="h5 fw-bold">Lampu Tidur</p>
  <div class="text-muted small">Small Size - Home</div>
  <div class="text-muted small">Stock: 13</div>
  <div class="d-flex justify-content-start">
    <p class="text-danger me-3"><s>Rp 28.000</s></p>
    <p class="fw-bold">Rp. 25.000</p>
  </div>

  <div class="fw-bold"> Anto Store</div>
  <div class="text-muted small"> Operational Hour : 10.00 - 22.00 WIB</div>
```

Gambar 3. 42 Coding informasi operational hour

Pada gambar 3.42 diatas merupakan coding front end untuk informasi operational pada halaman item detail. Penambahan coding hanya pada penambahan label saja. Terdapat label untuk operational hour dengan class "text-muted small" untuk membuat size font dari label kecil. Operational hour pada tampilan front end belum terhubung dengan data base sehingga masih menggunakan hard code. Berikut adalah hasil coding tampilan front end:



Gambar 3. 43 Hasil coding front end informasi operational hour

Pada fase ini, penulis mendapatkan tugas untuk membuat tampilan *front end* master sales. Pembuatan *mockup* telah dibuat oleh tim *system analyst* lain sehingga

penulis hanya perlu membuat tampilan *front end*. Pembuatan tampilan *front end* mengikuti dokumen spesifikasi program (SP). Berikut *coding front end* tampilan master sales:

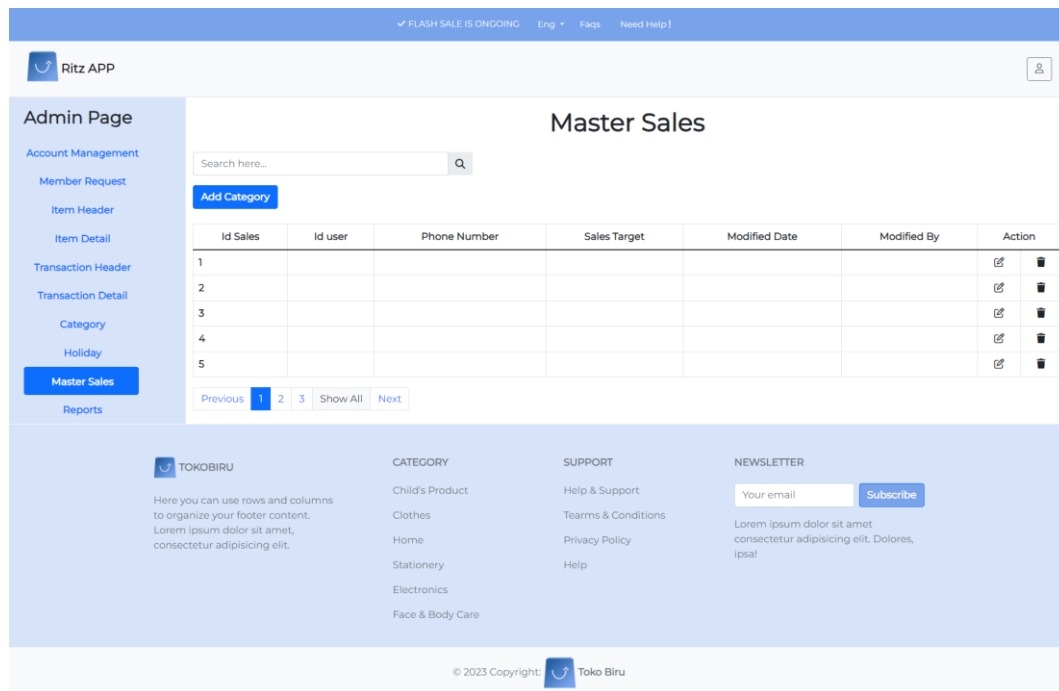
```

<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Master Sales</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." aria-label="Search">
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
      <button type="button" class="btn btn-primary mt-3"><b>Insert data</b></button>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">Id Sales</th>
            <th scope="col">Id user</th>
            <th scope="col">Phone Number</th>
            <th scope="col">Sales Target</th>
            <th scope="col">Modified Date</th>
            <th scope="col">Modified By</th>
            <th scope="col" colspan="2">Action</th>
          </tr>
        </thead>
      </table>
    </div>
  </main>
</div>

```

Gambar 3.44 Coding front end tampilan master sales

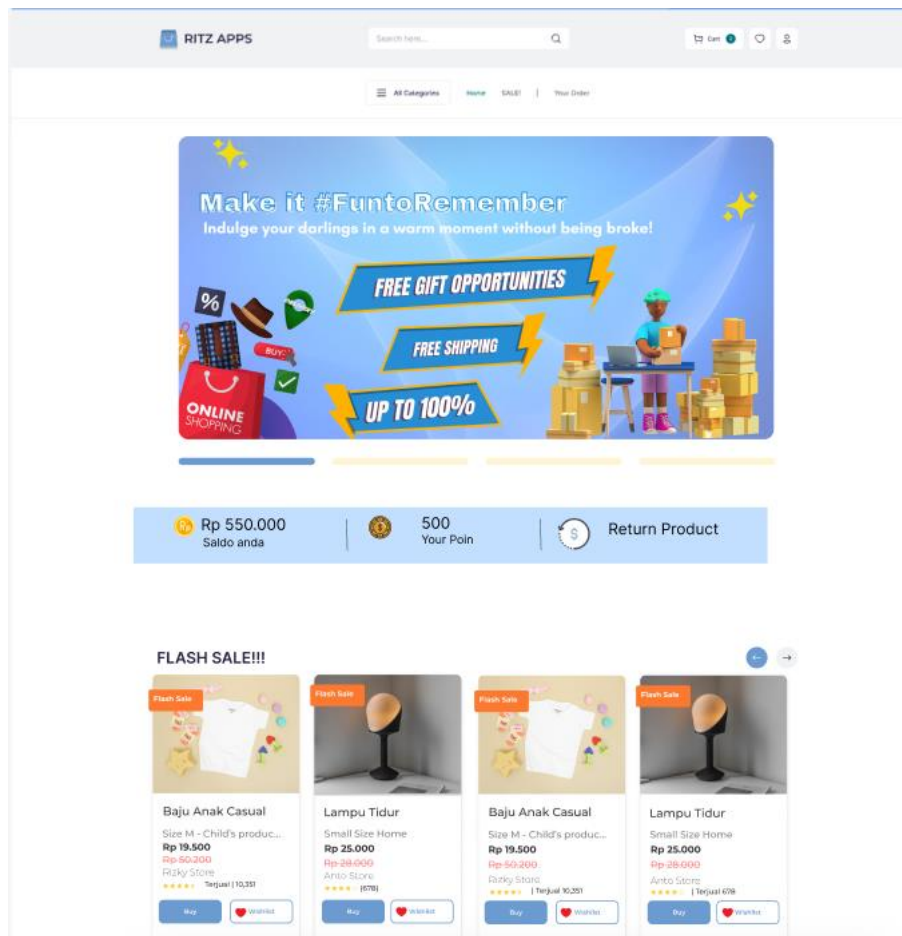
Pada gambar 3.44 diatas merupakan *front end coding* tampilan master sales. Terdapat label master sales dengan *class bold*. Untuk menambahkan data *discount* terdapat *button* dengan label “insert data”. Selanjutnya untuk membuat tabel menggunakan struktur tabel, ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat baris pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 7, salah satu tabel head digunakan untuk 2 *action button update, delete* sehingga kolom dibagi menjadi 2 bagian menggunakan *colspan*. Pada tabel *row <tr>* terdapat *class text center* agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada bagian bawah tabel terdapat *page navigation* yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukkan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan master sales:



Gambar 3. 45 Hasil coding front end tampilan master sales

10. pembuatan slider item pada tampilan *home page* pada 4 September 2023 – 9 September 2023

Pada fase ini, penulis akan mengerjakan slider item pada tampilan home page. Dalam satu bariss tampilan *home* maksimal hanya bisa memuat 4 item card. Slider berfungsi untuk menampilkan item card dalam satu bariss yang sama melebihi batas maksimal jumlah card item dalam satu bariss. Cara penggunaan slider dengan cara di geser ke kiri atau kanan. Pembuatan slider dilakukan pada halaman home page mengikuti mockup dokumen spesifikasi program. *Mockup* penempatan slider telah di buat oleh batch pertama, serta penempatan slider pada tampilan *front end home page* telah di buat namun slider tidak berfungsi karena hanya sekedar template. Berikut adalah mockup *UI/UX* tampilan homepage:



Gambar 3. 46 Mockup penempatan slider pada tampilan home

Penulis mendapatkan kebebasan oleh supervisor untuk memilih jenis *slider* yang akan di gunakan. Pada fase ini penulis belum memiliki pengalaman dan pengetahuan untuk membuat slider item card, sehingga penulis harus mempelajari dan mencari referensi slider. Pada tahap ini, Pembuatan slider masih bersifat eksperimental sehingga penulis membuat slider tidak pada file front end home page agar tidak mengganggu susunan coding dan hasil tampilan, penulis membuat file khusus hanya untuk slider saja. Setelah mempelajari dan mencari referensi slider, penulis menggunakan slider owl carousel. Berikut adalah proses pembuatan slider:

```

<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="
    <!-- Owl Carousel CSS -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.cs
    integrity="sha512-tS3S5qG0B1hnrQOyJXvNjeEM4UpMXHrQfTGmbQ1gKme1Cx1SEBUaxhRBj/EFTzpbP4RVSrpEikbmdJobCvhE3g=" cross
    referencerpolicy="no-referrer" />

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.theme.default.m
    integrity="sha512-sMxTmN1zRz0lHYKEujM2AqCLUR9F2C4/05cdbxjJLSRVMQIciEPCQZo+nk7go3BTsUK9kfa/s+a4f4i5pLkw=" cross
    referencerpolicy="no-referrer" />
    <title>Hello, world!</title>
  </head>

```

Gambar 3. 47 Import Bootstrap, Owl carousel CSS

Pada gambar 3.47 di atas merupakan import coding untuk CSS. Langkah pertama adalah melakukan import Bootstrap dan Carousel CSS untuk template tampilan dan juga membuat tampilan lebih responsive. Pembuatan slider menggunakan HTML, CSS, dan javascript.

```

<h1 class="text-center">Flash Sale!!!</h1>
  <div class="container">
    <div class="row">
      <div class="owl-carousel owl-theme">
        <div class="card w-100 shadow" style="width: 18rem">
          
          <div class="card-body">
            <h5 class="card-title fw-bold">Lampu Tidur 1</h5>
            <p class="text-muted card-text">Small Size Home</p>
            
            <div class="row">
              <div class="col-md">
                <p class="text-danger"><s>Rp 28.000</s></p>
              </div>
              <div class="col-md text-end me-5">
                <p class="fw-bold">Rp. 25.000</p>
              </div>
            </div>
            <div class="row">
              <div class="col-md">
                <button
                  type="button"
                  class="btn btn-outline-primary d-block w-100 text-white"
                  style="background-color: #6b9bd0"
                >
                  Buy
                </button>
              </div>
            </div>

```

Gambar 3. 48 Coding front end slider card

Pada gambar 3.48 merupakan coding untuk slider card. Terdapat class container untuk sebagai pembungkus card item. Selanjutnya di dalam class container terdapat class "card w-100 shadow" style="width: 18rem" untuk membuat card item serta mengatur ukuran. Kemudian terdapat untuk memasukan gambar pada card item, terdapat class card body fungsinya untuk membungkus isi dari card item. Di dalam card item terdapat nama barang, category, nama toko, harga barang, dan dua button action untuk buy, wishlist.

```

<!--JQUERY-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.0/jquery.min.js"
integrity="sha512-3gJwYpMe3QewGELv8k/BX9VvcqhrYRdzRMxVfq6ngyWXwo03GFEzjsUm8Q7RZcHPHkstattq7/GFoxjCVUjkjvPdw=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>

<!--Java Script Owl-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/owl.carousel.min.js"
integrity="sha512-bPs7Ae6pVvh0SiIcyUC1R7/q20AsRiovw4vAKX+zJbw3ShAeeqezq50RIIcIURq70a20rW2n2q+fyXBNcU91rw=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>

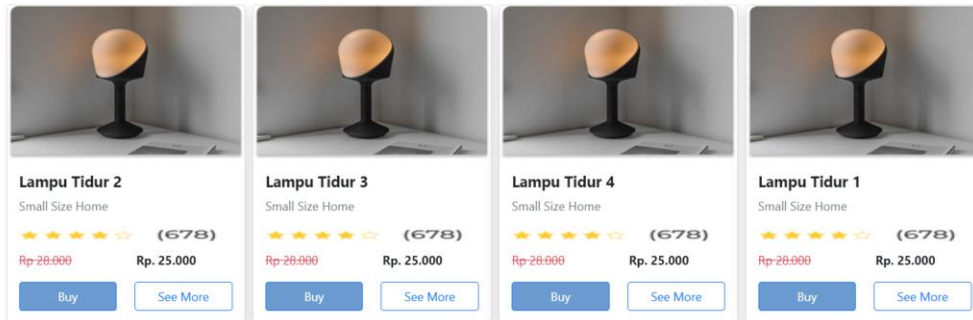
<script>
$( '.owl-carousel' ).owlCarousel({
  loop:true,
  margin:10,
  nav:true,
  responsive:{
    0:{
      items:3
    },
    600:{
      items:3
    },
    1000:{
      items :4
    }
  }
})
</script>

```

Gambar 3. 49 Javascript Owl carousel

Pada gambar 3.49 merupakan import Javascript owl carousel. Terdapat dua import javascript yang pertama import script untuk membuat slider owl carousel berfungsi, import script dalam bentuk link, kedua import script untuk cara kerja slider dan responsive. Pada script terdapat function loop dengan pernyataan true yang artinya slider akan melakukan pengulangan, jika item card sudah sampai akhir akan kembali ke item card pertama, function "margin:10" untuk mengatur jarak card item sebesar 10 pixel, lalu terdapat function responsive agar tampilan slider tetap bisa di gunakan dalam setiap ukuran. Terdapat 3 ukuran responsive jika normal akan muncul 4 card item, sedang muncul 3 card item item, dan terkecil 3 card item. Berikut adalah hasil coding slider owl carousel dari berbagai ukuran:

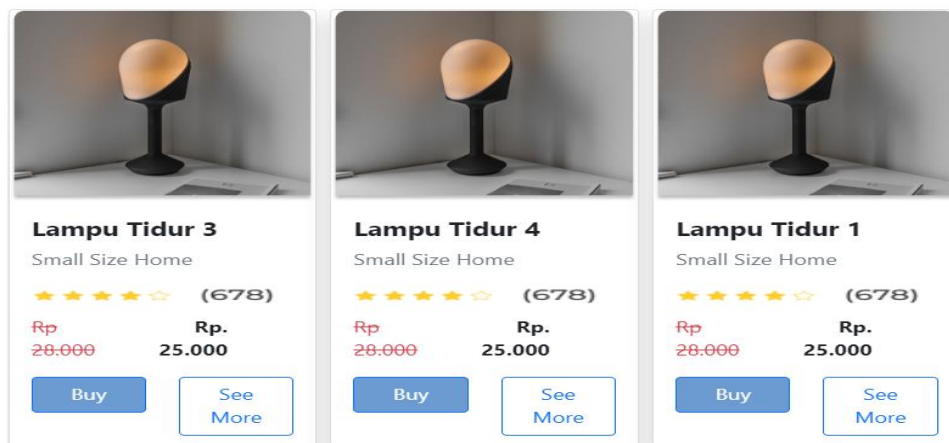
Flash Sale!!!



Gambar 3.50 Hasil coding front end slider

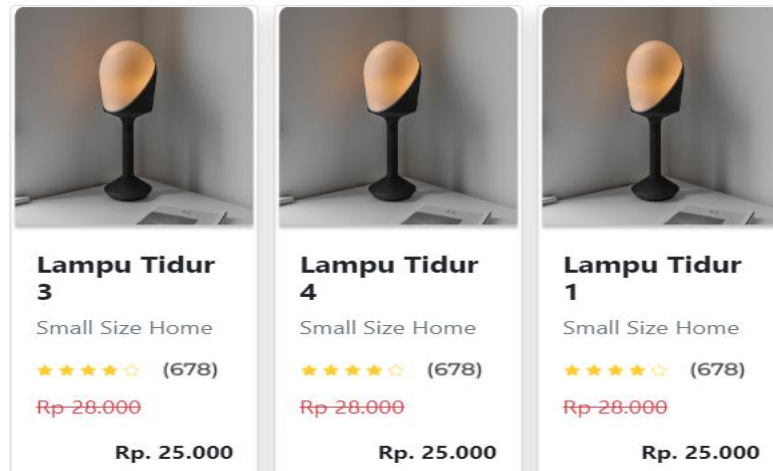
Pada gambar 3.50 di atas merupakan hasil slider owl carousel dengan ukuran responsive normal. Terlihat function loop dari card item lampu tidur 4 kembali ke card item lampu tidur 1.

Flash Sale!!!



Gambar 3.51 Hasil coding front end slider responsive sedang

Pada 3.51 gambar di atas merupakan hasil slider owl carousel dengan ukuran responsive sedang. Terlihat function loop dari card item lampu tidur 4 kembali ke card item lampu tidur 1.



Gambar 3. 52 Hasil coding front end slider responsive terkecil

Pada gambar 3.52 di atas merupakan hasil slider owl carousel dengan ukuran responsive terkecil. Terlihat function loop dari card item lampu tidur 4 kembali ke card item lampu tidur 1.

11. Revisi slider card pada tampilan *home page*, dan Pembuatan activity diagram fitur saldo pada tanggal 11 September – 16 September 2023

Pada tahap ini, penulis telah menunjukkan slider item menggunakan owl carousel kepada supervisor, akan tetapi supervisor meminta revisi untuk cara kerja dari slider. Owl carousel memiliki cara kerja dengan cara di *swipe* atau geser ke kanan atau kiri. Supervisor menginginkan cara kerja slider dengan cara button *previous* dan *next*. Penulis merasa pembuatan button *previous* dan *next* untuk owl carousel terlalu kompleks sehingga penulis mengganti slider menggunakan slick slider. Pembuatan slick slider bersifat eksperimental atau uji coba sehingga penulis tidak langsung membuat coding front end slider pada tampilan home, penulis membuat file khusus untuk slider saja. Berikut adalah proses coding pembuatan slider item menggunakan slick slider:


```

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sh
  <!-- SlickCSS -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.css" integrity="sha5
  <!-- Import Font Awesome -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" integrity="
  <title>Slick Slider</title>
</head>

```

Gambar 3. 53 Import CSS

Pada 3.53 gambar diatas merupakan coding untuk import CSS untuk pembuatan slider. Terdapat 3 item css yang di import yaitu: bootstrap css sebagai template halaman website, Slick css untuk template slider, dan Font awesome untuk mengimport ikon yang nantinya akan digunakan pada *previous* dan *next button*. Pembuatan slider menggunakan HTML, CSS, dan Javascript.

```

<div class="multiple-items">
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title fw-bold">Lampu Tidur 1</h5>
      <p class="text-muted card-text">Small Size Home</p>
      
      <div class="row">
        <div class="col-md">
          <p class="text-danger"><s>Rp 28.000</s></p>
        </div>
        <div class="col-md text-end me-5">
          <p class="fw-bold">Rp. 25.000</p>
        </div>
      </div>
      <div class="row">
        <div class="col-md">
          <button
            type="button"
            class="btn btn-outline-primary d-block w-100 text-white"
            style="background-color: #6b9bd0"
          >
            Buy
          </button>
        </div>
        <div class="col-md text-end">
          <button
            type="button"

```

Gambar 3. 54 Coding front end slick slider

Pada gambar 3.54 merupakan coding untuk slider card. Terdapat class "multiple items" sebagai pembungkus card item. Selanjutnya di dalam class container terdapat class *card* untuk membuat card item serta mengatur ukuran. Kemudian terdapat `` untuk memasukan gambar pada card item, terdapat class card body fungsinya untuk membungkus isi dari card item. Di dalam card item terdapat nama barang, category, nama toko, harga barang, dan dua button action untuk buy, wishlist.

```
<!--SlickSliderJS-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.min.js" integrity="sha512-XtmMtDEcNz
<script type="text/JavaScript">
    $('.multiple-items').slick({
        infinite: true,
        arrows : true,
        slidesToShow: 4,
        slidesToScroll: 4,
        prevArrow : '<span class="prev_arrow"><i class="fa-solid fa-arrow-left"></i></span>',
        nextArrow : '<span class="next_arrow"><i class="fa-solid fa-arrow-right"></i></span>',
    });
</script>d
```

Gambar 3. 55 Javascript Slick slider

Pada gambar 3.55 merupakan *import* Javascript Slick slider. Terdapat dua import javascript yang pertama *import script* untuk membuat slider *slick* slide berfungsi, *import script* dalam bentuk link, kedua import script untuk cara kerja slider dan *responsive*. Pada script terdapat *function infinite* dengan pernyataan *true* yang artinya slider akan melakukan pengulangan, jika *item card* sudah sampai akhir akan kembali ke *item card* pertama, *function arrow* dengan pernyataan *true* berfungsi untuk mengaktifkan fungsi *previous* dan *next button*, *slide to show* dengan pertanyaan empat maksudnya dalam satu kali *slide* terdapat empat card item yang akan bergeser. *prevArrow* dan *nextArrow* berfungsi untuk menghubungkan *button* dengan *slider*, serta untuk mengganti *button* dengan ikon.

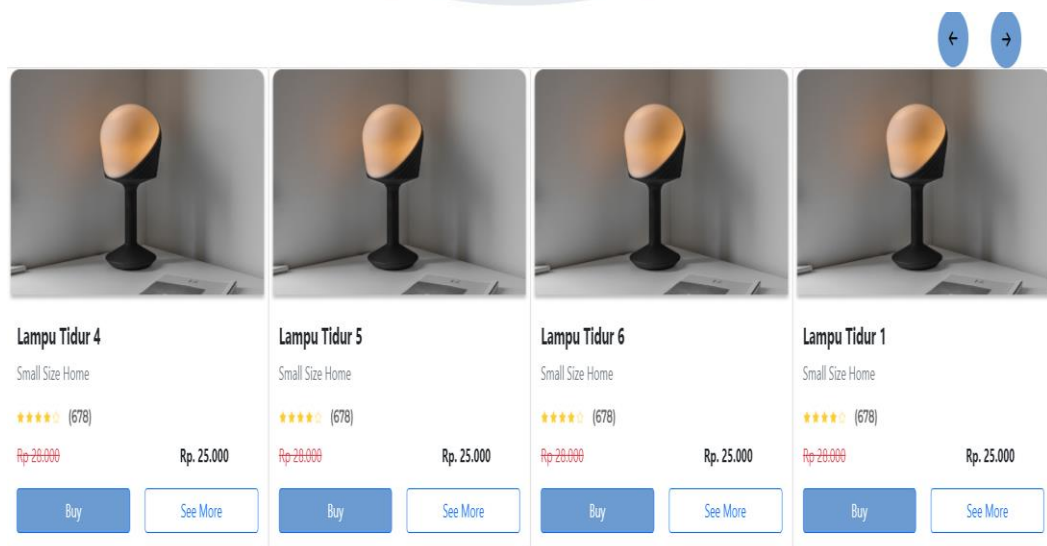
UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
<style type="text/css">
.prev_arrow{
  position:relative;
  left: 89%;
  color: black;
  width: 50px;
  height: 50px;
  background: #6b9bd0;
  border-radius: 50%;
  text-align: center;
  line-height: 50px;
  display: inline-block;
}

.next_arrow{
  position: absolute;
  margin-top: -463px;
  right: 3%;
  color: black;
  width: 50px;
  height: 50px;
  background: #6b9bd0;
  border-radius: 50%;
  text-align: center;
  line-height: 50px;
  display: inline-block;
}
</style>
```

Gambar 3. 56 CSS button previous dan next

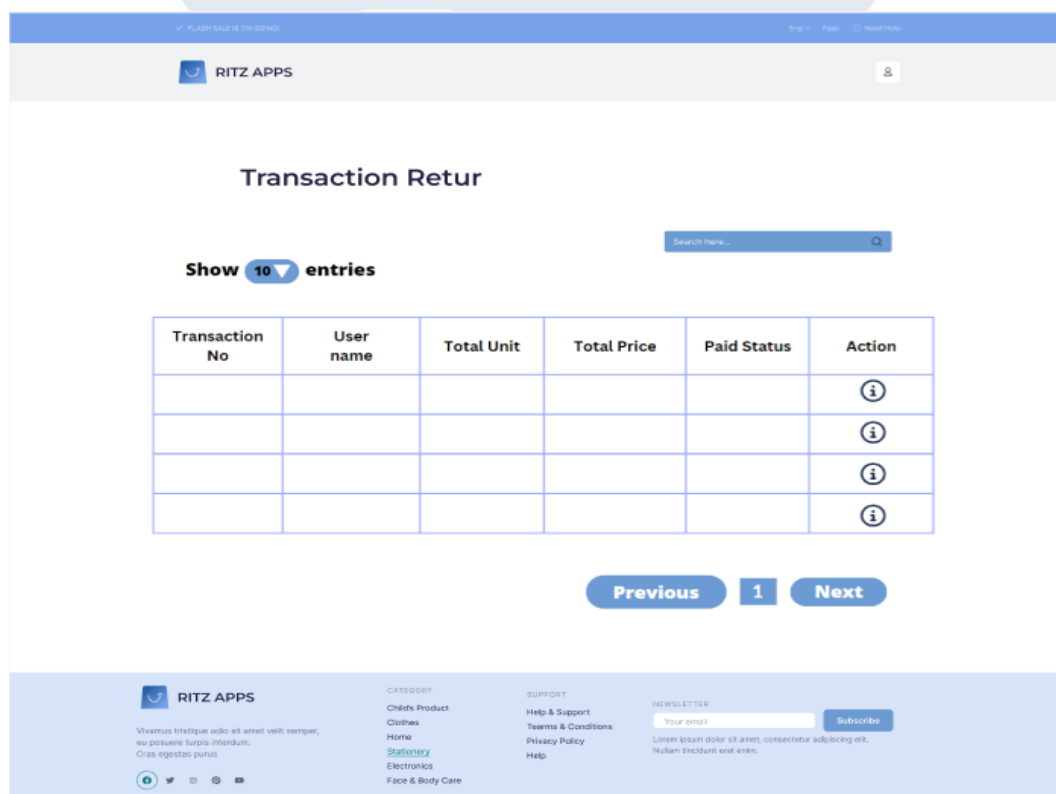
Pada gambar 3.56 diatas merupakan CSS untuk mengatur posisi dan bentuk dari button next dan previous button. Berikut adalah hasil keseluruhan coding front end slick slider:



Gambar 3. 57 hasil coding front end slick slider

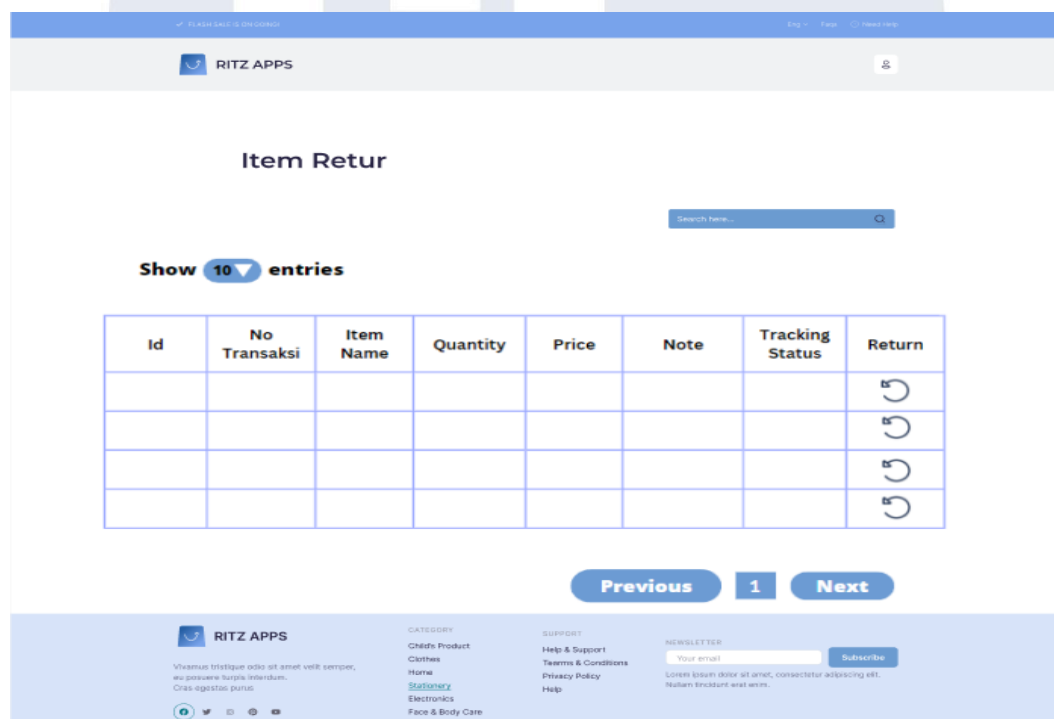
12. Pembuatan mockup fitur retur sisi user dan admin menggunakan figma pada tanggal 18 September 2023 – 23 September 2023

Pada fase ini penulis membuat mockup untuk fitur refund. Fitur retur adalah fitur user untuk pengembalian barang. Alur user untuk retur item yaitu pertama user akan masuk kedalam tampilan retur transaksi kemudian user memilih transaksi, kemudian user akan masuk ke dalam tampilan transaksi detail, pada tampilan tersebut user bisa memilih item mana yang ingin di retur. Tampilan retur untuk admin berfungsi untuk menerima atau menolak *request* return item dari user. Terdapat tiga mockup refund yang akan dibuat, dua mockup retur untuk sisi user, dan satu untuk sisi admin. Mockup dibuat menggunakan aplikasi figma. Berikut adalah mockup untuk fitur retur:



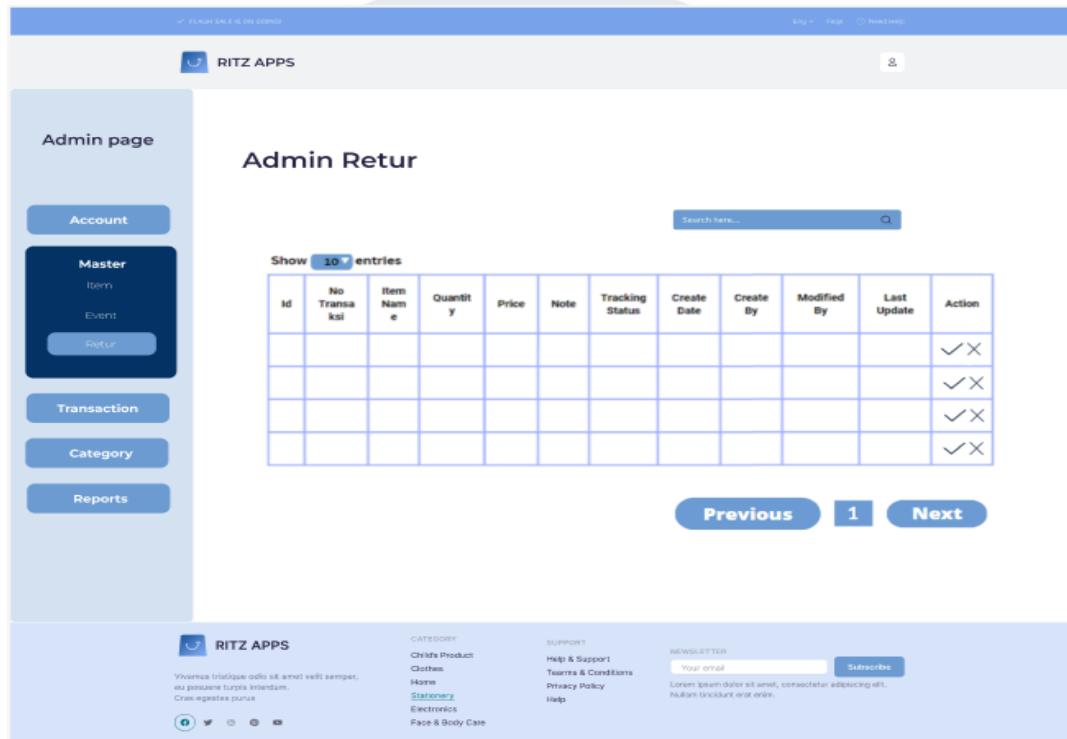
Gambar 3. 58 Mockup Transaction return

Pada gambar 3.58 merupakan mockup transaction return yang di buat menggunakan figma. Pada tampilan terdapat search bar untuk mencari transaksi. Terdapat tabel untuk menyimpan data transaksi yang telah dilakukan. Terdapat enam kolom yaitu Transaction no, Username, Total uni, Total price, Paid status dan action. Terdapat ikon dengan bentuk information untuk masuk ke tampilan item return. Terdapat page navigation untuk berpindah halaman tabel.



Gambar 3. 59 Mockup item return

Pada gambar 3.59 merupakan mockup item return yang di buat menggunakan figma. Pada tampilan terdapat search bar untuk mencari item. Terdapat tabel untuk memunculkan item yang ada dalam transaksi tersebut. Terdapat delapan kolom yaitu Id, No transaksi, Item name, Quantity, Price, Note, Tracking status dan action. Terdapat ikon untuk melakukan action return. Terdapat page navigation untuk berpindah halaman tabel.



Gambar 3. 60 Mockup return admin

Pada gambar 3.60 merupakan mockup master return yang di buat menggunakan figma. Pada tampilan tersebut terdapat *side bar* untuk memilih tampilan. kemudian terdapat search bar untuk mencari data pada tabel. Terdapat tabel untuk menampilkan *request return item* dari user, ada 12 kolom yaitu id, No transaction, item name, quantity, price, note, tracking status, create date, create by, modified by, last update. dan action. Terdapat dua ikon untuk *accept* dan *decline*. Terdapat page navigation untuk berpindah halaman tabel.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

13. Pembuatan front end tampilan fitur retur sisi user dan admin pada tanggal 25 September 2023 – 30 september 2023

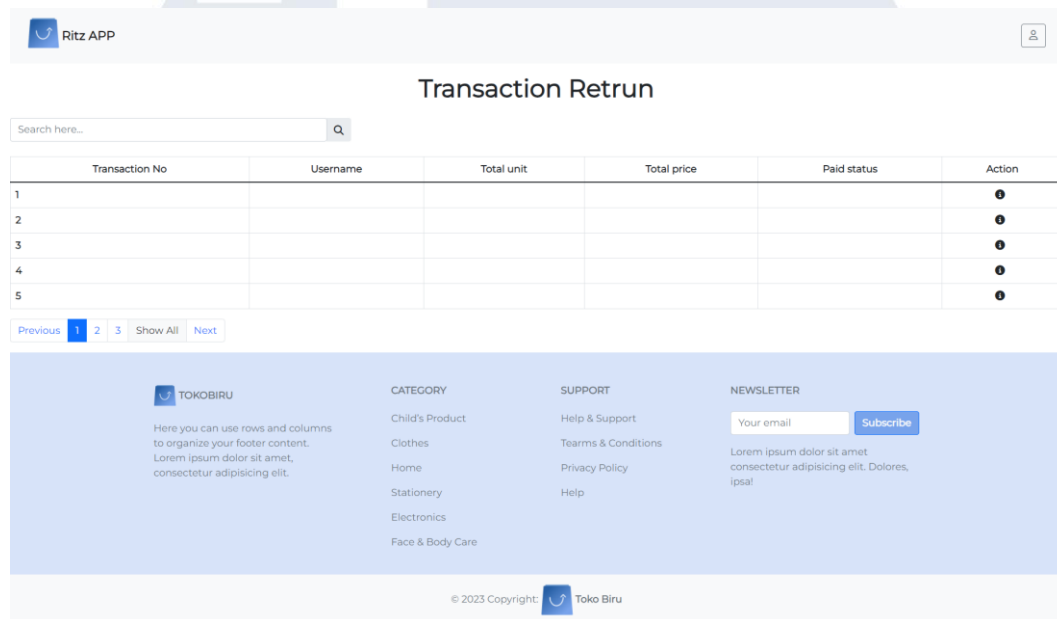
Setelah membuat mockup untuk fitur retur, penulis membuat tampilan front end fitur return. Pembuatan front end mengikuti mockup yang telah dibuat. Tampilan di buat menggunakan HTML, CSS. Tampilan belum terhubung dengan database karena merupakan pekerjaan untuk sisi *back end*. Berikut coding tampilan front end fitur retur:

```
<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Transaction Retrun</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." aria-l:
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">Transaction No</th>
            <th scope="col">Username</th>
            <th scope="col">Total unit</th>
            <th scope="col">Total price</th>
            <th scope="col">Paid status</th>
            <th scope="col">Action</th>
          </tr>
        </thead>
      </table>
    </div>
  </main>
</div>
```

Gambar 3. 61 Coding front end transacstion retur

Pada gambar 3.61 diatas merupakan *front end coding* tampilan transaction return. Terdapat label transaction return sebagai judul halaman dengan *class bold*. Selanjutnya *coding* untuk membuat tabel menggunakan struktur tabel, ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat baris pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 6, salah satu tabel head digunakan untuk *action*. Icon action menggunakan `<i class="fa fa-info-circle me-2"></i>` yang di import dari font awesome. Pada tabel row `<tr>` terdapat class *text center* agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada

bagian bawah tabel terdapat page navigation yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukkan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan transaction return:



Gambar 3. 62 Hasil coding front end transaction return

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

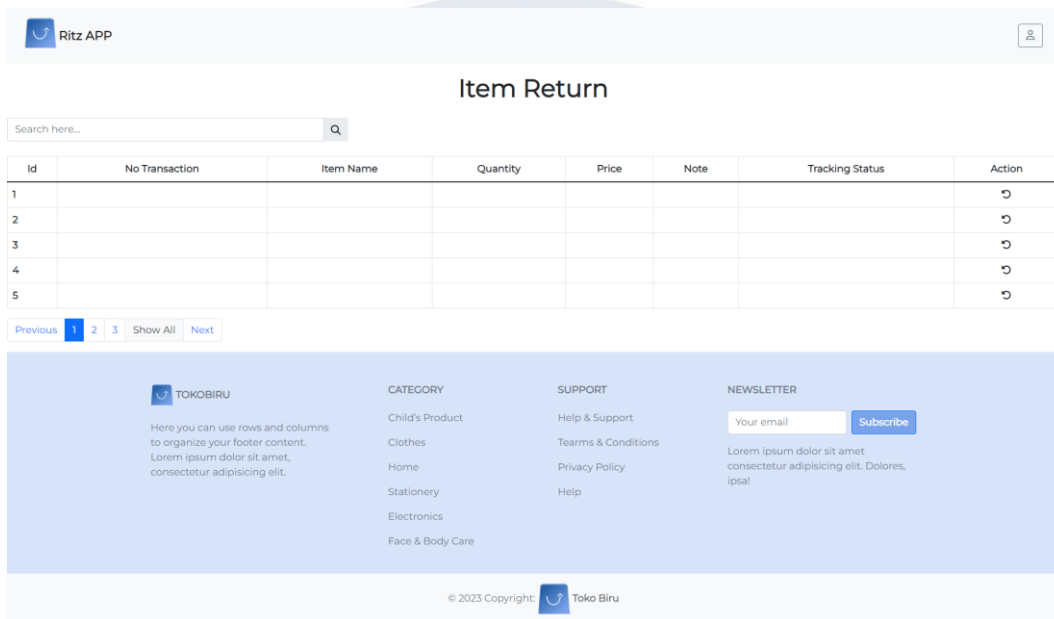

```

<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Item Return</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search"
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">Id</th>
            <th scope="col">No Transaction</th>
            <th scope="col">Item Name</th>
            <th scope="col">Quantity</th>
            <th scope="col">Price</th>
            <th scope="col">Note</th>
            <th scope="col">Tracking Status</th>
            <th scope="col">Action</th>
          </tr>

```

Gambar 3. 63 Coding front end tampilan item return

Pada gambar 3.63 diatas merupakan *front end coding* tampilan item return. Terdapat label item return sebagai judul halaman dengan *class bold*. Selanjutnya *coding* untuk membuat tabel menggunakan struktur tabel, ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat baris pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 8, salah satu tabel head digunakan untuk *action*. Icon action menggunakan `<i class="fa-solid fa-rotate-left"></i>` yang di import dari font awesome. Pada tabel row `<tr>` terdapat *class text center* agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada bagian bawah tabel terdapat page navigation yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukkan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan item return:



Gambar 3. 64 Hasil coding front end tampilan item return

```

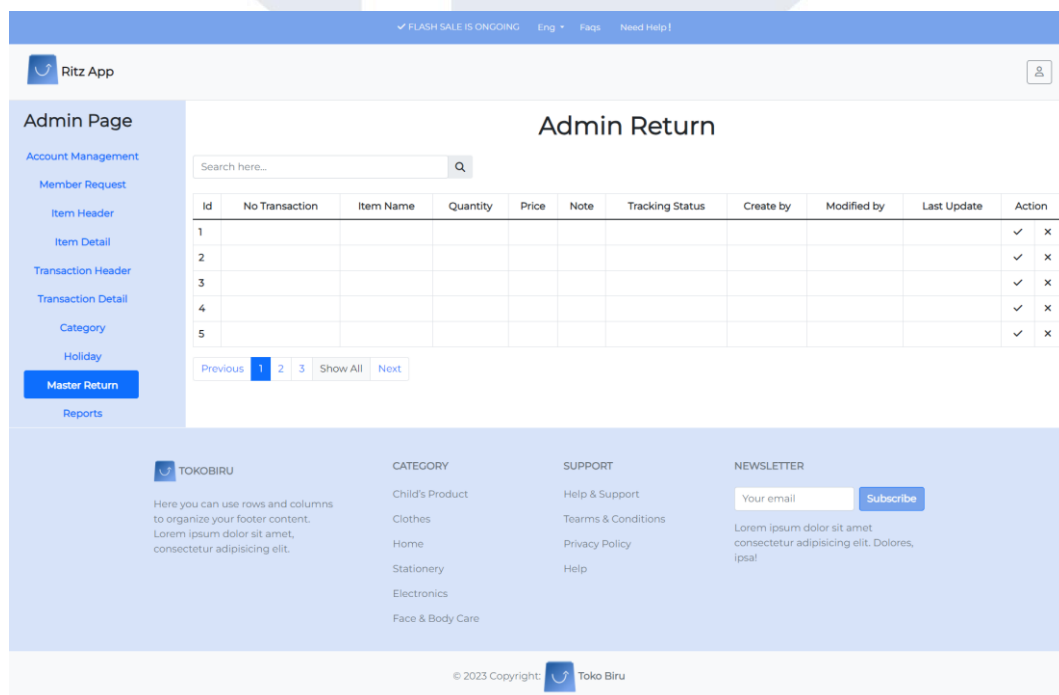
<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto">
    <h1 class="text-center mt-3 fw-bold">Admin Return</h1>
    <div class="col-md-4 mt-3">
      <div class="input-group rounded">
        <input type="search" class="form-control rounded" placeholder="Search here..." ar
        <span class="input-group-text border-0" id="search-addon">
          <i class="fa fa-search"></i>
        </span>
      </div>
    </div>
    <div class="col-md-12 pt-4">
      <table class="table table-bordered">
        <thead>
          <tr class="text-center">
            <th scope="col">Id</th>
            <th scope="col">No Transaction</th>
            <th scope="col">Item Name</th>
            <th scope="col">Quantity</th>
            <th scope="col">Price</th>
            <th scope="col">Note</th>
            <th scope="col">Tracking Status</th>
            <th scope="col">Create by</th>
            <th scope="col">Modified by</th>
            <th scope="col">Last Update</th>
            <th scope="col" colspan="2">Action</th>
          </tr>

```

Gambar 3. 65 Coding front end tampilan Master return admin

NUSANTARA

Pada gambar 3.65 diatas merupakan *front end coding* tampilan master return. Terdapat label Admin return sebagai judul tampilan halaman dengan *class bold*. Selanjutnya untuk membuat tabel menggunakan struktur tabel, ditambah dengan *class table border* untuk membuat garis pada setiap sisi kolom dan sisi tabel. Kemudian terdapat tabel *row* dengan tanda `<tr>`, tujuannya untuk membuat bariss pada kolom, didalamnya terdapat tabel *head* dengan tanda `<th>`, tujuannya untuk membuat judul pada setiap tabel. Jumlah kolom yang terdapat pada tabel adalah 11, salah satu tabel head `<th>` digunakan untuk 2 *action button accept, decline* sehingga kolom dibagi menjadi 2 bagian menggunakan `colspan`. Pada tabel *row <tr>* terdapat *class text center* agar setiap judul pada `<th>` selalu berada di posisi tengah. Terdapat fitur *search bar* untuk mempermudah mencari data, pada bagian bawah tabel terdapat *page navigation* yang disertai dengan *previous* dan *next button* fungsinya untuk menunjukkan halaman tabel dan mengganti halaman tabel. Berikut merupakan hasil *coding* tampilan master return:



Gambar 3. 66 Hasil coding front end tampilan master return

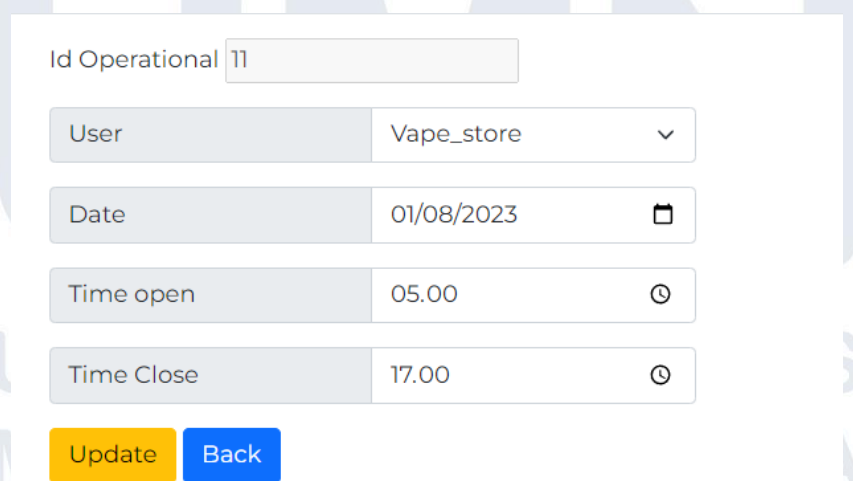
14. Coding penambahan back button pada tanggal 2 Oktober 2023 – 7 oktober 2023

Pada tahap ini, penulis menambahkan *button back*, fungsinya untuk kembali ke halaman sebelumnya. Tampilan yang mendapatkan tambahan *button back* terdapat pada tampilan action sisi admin seperti *update*, *delete*, *accept*, *decline*, dan lain lain. Penambahan coding button back dilakukan pada sisi back end dengan *framework* laravel dikarenakan penambahan *coding* yang sedikit dan tidak terlalu kompleks. Berikut *coding* penambahan *back button*:

```
<div class="form-group">
  <button type="submit" class="btn btn-warning">Update</button>
  <button type="button" onclick="history.back();" class="btn btn-primary">Back</button>
</div>
```

Gambar 3. 67 Coding button back pada tampilan action update master operational

Pada gambar 3.67 di atas merupakan *coding* untuk penambahan *back button*. Untuk membuat button bisa di klik terdapat `type="button"`. Terdapat `class="btn btn-primary"` untuk membuat *button* menggunakan warna utama yaitu biru, serta meningkatkan *awareness* pengguna. Pembuatan *button* pada sisi *back end* dengan *framework* laravel, terdapat coding `onclick="history.back()"` sebagai *function* untuk kembali ke halaman sebelumnya. *Coding back* pada gambar diatas digunakan pada tampilan lain. Berikut salah satu hasil tampilan coding back button pada tampilan action edit master operational:



Id Operational		11
User	Vape_store	▼
Date	01/08/2023	📅
Time open	05.00	🕒
Time Close	17.00	🕒
Update Back		

Gambar 3. 68 Hasil coding bak button

15. Membantu membuat skenario testing return item admin dan user, payterm, master item pada 9 Oktober 2023 – 21 Oktober 2023

Pada fase ini, penulis telah menyelesaikan task yang telah diberikan. Penulis menunggu feedback hasil testing QA. Untuk mengisi kekosongan dan mempercepat fase QA task penulis membantu membuat skenario testing. Tugas perbantuan ini telah dilaporkan dan disetujui oleh supervisor. Penulis membuat skenario testing untuk modul user return, payterm, dan modul transaction detail. Tabel skenario testing telah dibuat oleh QA sehingga penulis hanya membuat skenario. Berikut adalah skenario testing modul admin retur, payterm, dan modul user item detail:

Bagian	Test Scenario	Result Plan
localhost/UserPage/ReturnProduct/TransactionReturn	"Return" diklik untuk barang yang ingin di return	Maximal barang yang direturn 3 hari setelah sampai
localhost/UserPage/ReturnProduct	"Return Product" di klik pada halaman awal user	Masuk kedalam tampilan transaction return
localhost/UserPage/ReturnProduct/TransactionReturn	Button "Close" diklik di tampilan transaction return	Balik kedalam tampilan awal user
localhost/UserPage/ReturnProduct/TransactionReturn	Klik "Return item" untuk barang yang ingin di return	Masuk kedalam tampilan produk item
localhost/UserPage/ProductItem	Button "Back" diklik di tampilan produk return	Balik kedalam tampilan awal transaction return
localhost/UserPage/ProductItem	Button "Return" diklik	Masuk kedalam page return item
localhost/UserPage/ProductItem	Jumlah item diisi kecuali "Reason"	Muncul "Please fill out this field"
localhost/UserPage/ProductItem	Reason diisi kecuali "Jumlah item"	Muncul "Please fill out this field"
localhost/UserPage/ProductItem	Button "Back" diklik	Kembali ke product item
localhost/UserPage/ProductItem	Button "Save Changes" diklik	Adanya pesan "Success" dan kembali ketampilan awal user.

Gambar 3. 69 Skenario testing modul user retur

localhost/UserPage/Payterm	Button payterm di"klik"	masuk ke dalam tampilan tabel cicilan dalam setiap transaksi
localhost/UserPage/Payterm	button pay "diklik"	masuk ke tampilan pilihan pembayaran
localhost/UserPage/Payterm	"back to transaction displays" diklik	kembali ke halaman transaction details

Gambar 3. 70 skenario testing modul user payterm

localhost/transaction/Detail	Show entries dipilih "10"	Menampilkan 10 data pertama, sisanya dilanjutkan ke halaman berikutnya
localhost/transaction/Detail	Show entries dipilih "25"	Menampilkan 25 data pertama, sisanya dilanjutkan ke halaman berikutnya
localhost/transaction/Detail	Show entries dipilih "50"	Menampilkan 50 data pertama, sisanya dilanjutkan ke halaman berikutnya
localhost/transaction/Detail	Show entries dipilih "100"	Menampilkan 100 data pertama, sisanya dilanjutkan ke halaman berikutnya
localhost/transaction/Detail	"next" di klik	muncul halaman selanjutnya
localhost/transaction/Detail	"previous" di klik	kembali ke halaman sebelumnya
localhost/transaction/Detail	Mengisi "Bar Search" sesuai nama barang yang diinginkan	Menampilkan transaksi barang yang dicari

Gambar 3. 71 Skenario testing modul admin transaction detail

UNIVERSITAS
MULTIMEDIA
NUSANTARA

16. Membantu manual testing modul user return, payterm, dan modul user transaction detail pada tanggal 23 November 2023 – 4 November 2023

Pada tahap ini, setelah membuat skenario testing, penulis membantu manual testing untuk skenario yang telah di buat pada minggu lalu. Berikut adalah hasil manual testing mengikuti skenario yang sudah di buat:

Bagian	Test Scenario	Result Plan	Tanggal Testing	Actual Result	Kategori Hasil	PIC QA
localhost/UserPage/ReturnProduct/TransactionReturn	"Return" diklik untuk barang yang ingin di return	Maximal barang yang di return 3 hari setelah sampai	27/10/2023		PASSED	RICO
localhost/UserPage/ReturnProduct	"Return Product" di klik pada halaman awal user	Masuk kedalam tampilan transaction return	31/07/2023		PASSED	RICO
localhost/UserPage/ReturnProduct/TransactionReturn	Button "Close" diklik di tampilan transaction return	Balik kedalam tampilan awal user	31/07/2023		PASSED	RICO
localhost/UserPage/ReturnProduct/TransactionReturn	Klik "Return item" untuk barang yang ingin di return	Masuk kedalam tampilan produk item	31/07/2023		PASSED	RICO
localhost/UserPage/ProductItem	Button "Back" diklik di tampilan produk return	Balik kedalam tampilan awal transaction return	31/07/2023		PASSED	RICO
localhost/UserPage/ProductItem	Button "Return" diklik	Masuk kedalam page return item	31/07/2023		PASSED	RICO
localhost/UserPage/ProductItem	Jumlah item diisi kecuali "Reason"	Muncul "Please fill out this field"	31/07/2023		PASSED	RICO
localhost/UserPage/ProductItem	Reason diisi kecuali "Jumlah item"	Muncul "Please fill out this field"	31/07/2023		PASSED	RICO
localhost/UserPage/ProductItem	Button "Back" diklik	Kembali ke product item	31/07/2023		PASSED	RICO
localhost/UserPage/ProductItem	Button "Save Changes" diklik	Adanya pesan "Success" dan kembali ketampilan awal user.	31/07/2023		PASSED	RICO

Gambar 3. 72 Hasil manual testing pada modul user return

Pada gambar 3.72 diatas terdapat kesalahan input tanggal testing pada tabel. Testing dilakukan pada tanggal 27 November 2023.

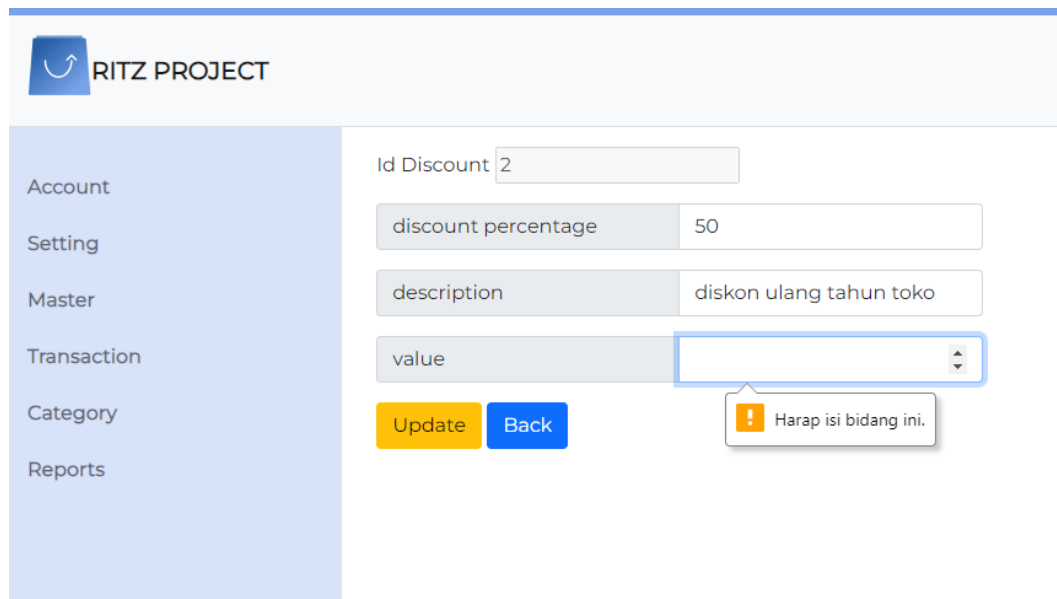
localhost/UserPage/Payterm	Button payterm di"klik"	masuk ke dalam tampilan tabel cicilan dalam setiap transaksi	31/07/2023		PASSED	RICO
localhost/UserPage/Payterm	button pay "diklik"	masuk ke tampilan pilihan pembayaran	31/07/2023		PASSED	RICO
localhost/UserPage/Payterm	"back to transaction displays" diklik	kembali ke halaman transaction details	31/07/2023		PASSED	RICO

Gambar 3. 73 Hasil manual testing modul user payterm

localhost/transaction/Detail	Show entries dipilih "10"	Menampilkan 10 data pertama, sisanya dilanjutkan ke halaman berikutnya			PASSED	RICO
localhost/transaction/Detail	Show entries dipilih "25"	Menampilkan 25 data pertama, sisanya dilanjutkan ke halaman berikutnya			PASSED	RICO
localhost/transaction/Detail	Show entries dipilih "50"	Menampilkan 50 data pertama, sisanya dilanjutkan ke halaman berikutnya			PASSED	RICO
localhost/transaction/Detail	Show entries dipilih "100"	Menampilkan 100 data pertama, sisanya dilanjutkan ke halaman berikutnya			PASSED	RICO
localhost/transaction/Detail	"next" di klik	muncul halaman selanjutnya			PASSED	RICO
localhost/transaction/Detail	"previous" di klik	kembali ke halaman sebelumnya			PASSED	RICO
localhost/transaction/Detail	Mengisi "Bar Search" sesuai nama barang yang diinginkan	Menampilkan transaksi barang yang dicari			PASSED	RICO

Gambar 3. 74 Manual testing modul admin transaction detail

Setelah melakukan testing dengan mengikuti skenario yang telah di buat, mendapatkan hasil seluruh skenario *passed* atau lulus *testing*. Pada saat melakukan *random* testing diluar skenario yang telah dibuat, penulis menemukan *error logic*. Error ditemukan pada tampilan action edit master discount.

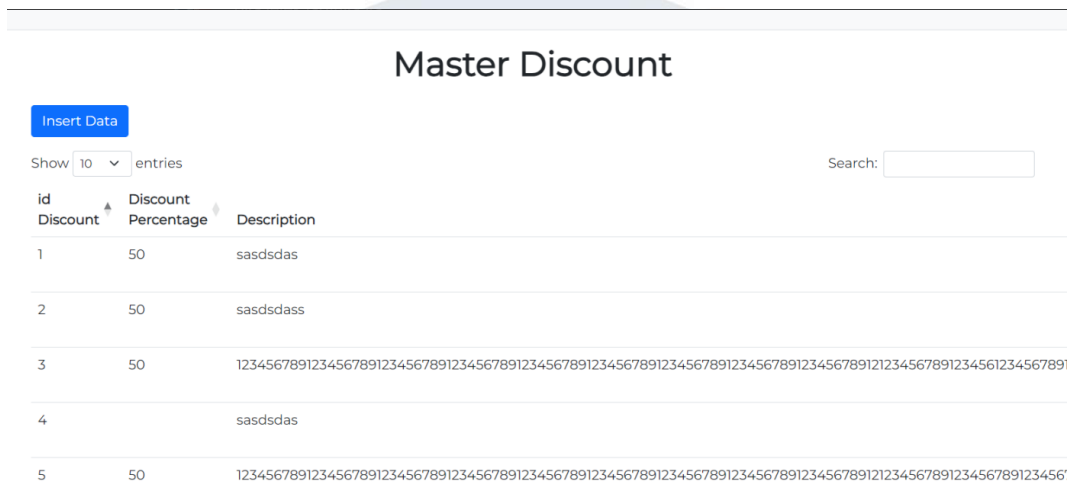


Gambar 3.75 Error logic action edit master discount

Pada gambar 3.75 di atas merupakan error logic pada bagian action edit master discount. Error logic terjadi pada saat edit pengisian diskon, kondisi atau *actual result* yang harusnya terjadi adalah hanya boleh satu tipe diskon yang dapat terisi antara discount percentage atau value, mengambil contoh dari gambar diatas jika salah satu input box discount percentage telah terisi maka input box value akan disabled/ tidak bisa di isi. Terdapat error logic dimana saat akan mengedit diskon ke dua input box tipe diskon harus terisi, jika tidak discount tidak bisa terupdate.

17. Fixing tampilan fail testing pada tanggal 6 November – 11 November 2023

Setelah testing website telah dilakukan oleh QA, penulis telah mendapatkan feedback result testing. Tim QA menyampaikan feedback terdapat fail testing pada tampilan tabel, fail testing yang terjadi adalah tampilan tabel akan melebar. Tabel melebar dikarenakan lebar kolom yang fleksibel, sehingga jika terdapat kalimat atau kata yang panjang akan terus melebar ke kanan, hal tersebut akan mempengaruhi besar kolom lainnya dan merusak estetika tabel. Berikut contoh tabel pada tampilan master discount:



Gambar 3. 76 Fail testing tampilan tabel

Pada gambar 3.76 diatas terlihat bahwa isi dari kolom description melebar kekanan. Fail tampilan tabel terjadi karena tidak ada ukuran *absolute* untuk lebar dari masing masing kolom. Penulis menambahkan coding untuk menentukan lebar dari setiap kolom. Berikut coding yang ditambahkan:

```

19
20 <div class="col d-flex flex-column h-sm-100">
21 <h1 class="text-center mt-3 fw-bold">Master Discount</h1>
22 <div class="col-md-4 mt-3 ps-5">
23 <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-ta
24 </div>
25 <div class="col-md ps-5 pe-5" style="padding-top:1rem;">
26 <table id="example" class="table table-striped" >
27 <thead>
28 <tr class="text-center">
29 <th style="width:100%">id Discount</th>
30 <th style="width:100%">Discount Percentage</th>
31 <th style="width:150%">Description</th>
32 <th style="width:50%">Price</th>
33 <th style="width:50%">Created by</th>
34 <th style="width:100%">Created Date</th>
35 <th style="width:50%">Modified by</th>
36 <th style="width:100%">Last Update</th>
37 <th style="width:50%">Action</th>
38 </tr>
39 </thead>

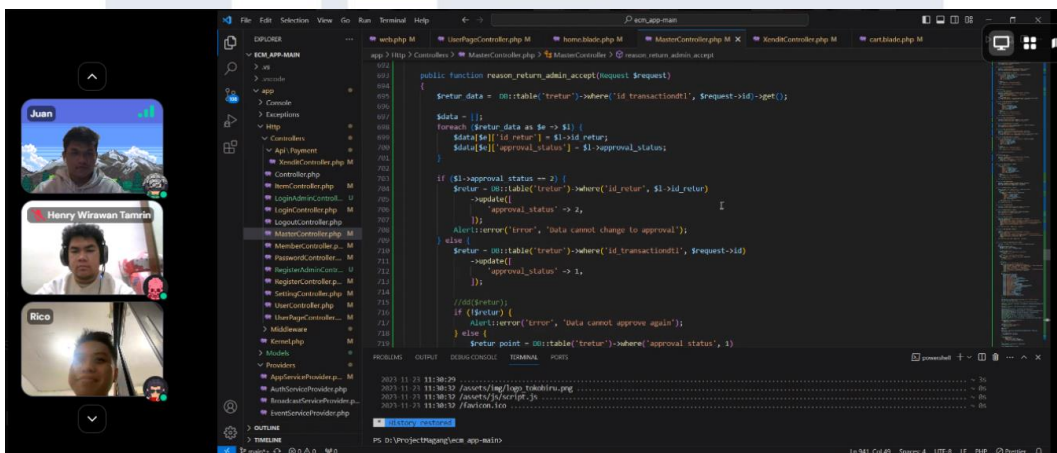
```

Gambar 3. 77 Update coding lebar kolom table

Pada gambar 3.77 diatas penulis menambahkan coding untuk menentukan lebar dari setiap kolom. Penambahan *style* pada table head <th> membuat lebar



Gambar 3. 80 Dokumentasi Phase Feedback UAT Result



Gambar 3. 81 Dokumentasi Phase Feedback UAT Result

3.3 Kendala yang ditemukan

Selama pelaksanaan magang yang penulis telah dilalukan sebagai *System analyst* dan *Front end* pada PT. Ritzproject, terdapat beberapa kendala yang ditemukan yaitu:

1. Pada fase awal magang penulis tidak mengetahui perbedaan file yang di berikan oleh tim *front end* dan *back end* pemegang batch pertama. Dalam pembuatan coding tampilan penulis tidak mengetahui perbedaan file untuk tim *front end* dan *back end*.
2. Koneksi *internet* rumah yang buruk. Pelaksanaan magang yang dilakukan secara *online* sangat membutuhkan koneksi *internet*, namun sering kali terjadi koneksi *internet* yang tiba-tiba memburuk. Hal tersebut mengganggu penulis dalam

melaksanakan kerja magang dikarenakan membutuhkan waktu yang lama untuk mengunduh file yang dibutuhkan dan kesulitan untuk berkomunikasi dengan tim.

3. Keterlambatan penyelesaian *project*. pekerjaan *project* dilakukan secara paralel, sehingga ketika ada keterlambatan penyelesaian *task* dari salah satu divisi akan berdampak ke waktu penyelesaian *task* divisi lain. Keterlambatan penyelesaian *project* membuat waktu yang dimiliki divisi QA untuk melakukan testing semakin sedikit.

3.4 Solusi atas kendala yang ditemukan

Solusi dari kendala yang telah di sebutkan pada poin sebelumnya yaitu:

1. mencari informasi secara langsung dengan cara bertanya kepada tim front end dan back-end batch pertama, serta secara mandiri mencari dan mempelajari perbedaanya melalui *internet*
2. untuk mempermudah komunikasi dengan tim penulis mengajak tim magang untuk bertemu secara tatap muka, bila membutuhkan komunikasi secara jarak jauh menggunakan pesan melalui aplikasi *chatting*. Dalam mengunduh *file* yang dibutuhkan penulis akan mencari koneksi *internet* yang lebih cepat dan stabil seperti *café* atau rumah teman.
3. Membantu mencari masalah dari *task* yang sedang di kerjakan oleh tim divisi magang. Untuk mempercepat QA melakukan testing penulis memberikan bantuan testing.