

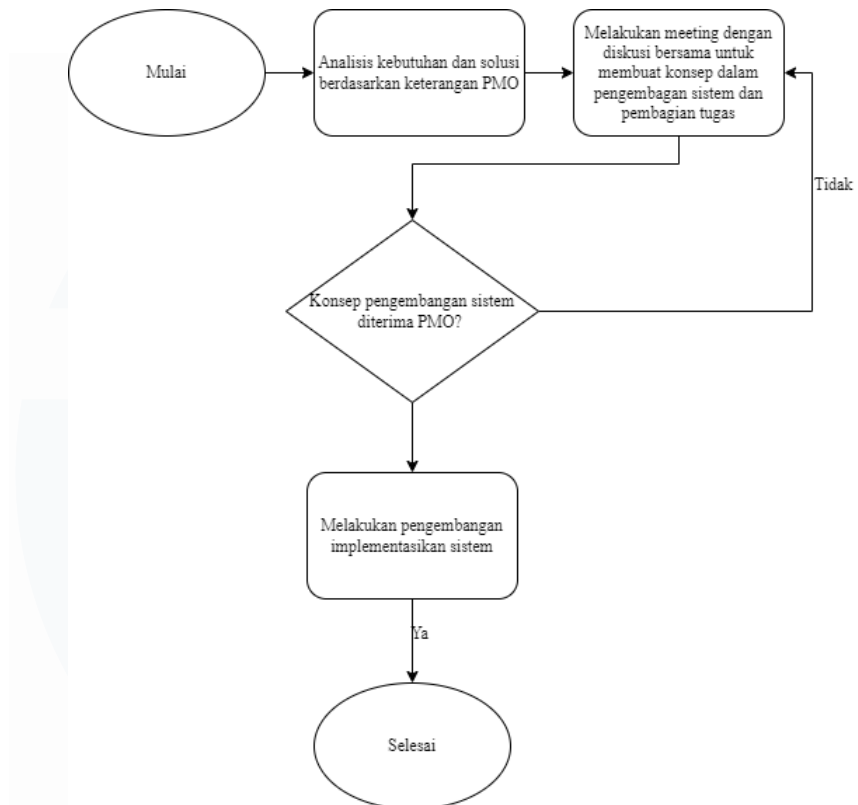
BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Penulis terlibat dalam kegiatan magang di PT Ritzproject sebagai pengembang perangkat lunak (*Back-end*). Dalam perannya sebagai *Back-end*, penulis melakukan pengkodean sesuai dengan petunjuk tugas yang diberikan oleh *System Analyst*. Tugas tersebut melibatkan pengembangan sistem untuk website *e-commerce*, yang merupakan buatan dari PT Ritzproject. Pertamanya, website *e-commerce* direncanakan menjadi produk internal PT Ritzproject. Namun, dalam proses pengembangan, ada *client* yang menunjukkan minat untuk memesan sebuah sistem tersebut. Penulis bersama tim bertanggung jawab pada pembuatan *template* website *e-commerce* yang bersifat universal.

Sewaktu menjalani periode magang, penulis mendapatkan bimbingan langsung dari Bapak Farid Ananda, yang menjabat sebagai *Project Management Office* (PMO) di PT Ritzproject. Di bawah bimbingannya, penulis menerima petunjuk serta dukungan saat terlibat dalam pengembangan sistem *e-commerce*. Penulis memiliki tim terdiri dari *System analyst*, *Developer software*, dan *Quality assurance*, dengan total anggota tim sebanyak 8 orang. Tim ini terdiri dari satu *System analyst* (*website*), satu *Developer* (*front-end*), lima *Developer* (*back-end*), dan satu *Quality Assurance*. Penulis menempati peran sebagai pengembang perangkat lunak (*back-end*) dalam tim. Bersama-sama, tim berhasil menciptakan Produk yang ditawarkan oleh PT Ritzproject mencakup sistem *e-commerce*. Meskipun proyek berhasil menarik perhatian klien selama proses pengerjaan, tim penulis hanya bertugas membuat halaman sebagai bagian dari produk internal PT Ritzproject. Penyesuaian sesuai keinginan klien untuk dihandle oleh karyawan tetap PT Ritzproject.



Gambar 3.1 Proses operasional PT Ritzproject Sinergi Visitama

Gambar 3.1 Penulis mencerminkan langkah-langkah yang diambil selama menjalani magang di PT Ritzproject. Fokus utama dari alur kerja ini terletak pada analisis kebutuhan dan penyelesaian masalah untuk merumuskan solusi yang memanfaatkan teknologi. Dalam konteks ini, peran *Project Management Office* (PMO), yang juga bertindak sebagai supervisor penulis selama magang, menjadi krusial karena setiap sistem yang dikembangkan harus mendapatkan persetujuan dari PMO. Sehingga, penulis dengan timnya terlibat dalam diskusi untuk mengidentifikasi mengembangkan metode serta melakukan pembagian *task* berdasarkan peran masing-masing.

Dalam forum diskusi yang membahas semua aspek terkait pengembangan sistem, partisipasi tidak hanya terbatas pada bagian tim, namun PMO juga seringkali terlibat dalam memberikan masukan dan arahan yang mendukung perkembangan sistem website *e-commerce*. Setelah

mendapat persetujuan dari PMO, tim kami akan memulai implementasi sistem *e-commerce* sesuai dengan tugas masing-masing. Penulis juga akan mengadakan pertemuan *progress report* setiap satu atau dua minggu bersama PMO PT Ritzproject untuk dibahas kemajuan yang telah dicapai.

3.2 Tugas dan Uraian Kerja Magang

Semasa menjalani program magang, dalam sebuah proyek untuk merancang dan memajukan mendasarkan *e-commerce* berbasis website. Pengembangan sistem *e-commerce* dapat dilakukan mulai pada tanggal 26 Juni 2023 hingga 15 Desember 2023. Semenjak memulai tahap pengembangan, akan melibatkan pemberian seluruh kebutuhan dan petunjuk oleh PMO PT Ritzproject. Setelah semua persiapan ini selesai, tim penulis dapat memulai pengerjaan proyek. Berikut adalah jadwal waktu yang diikuti selama menjalani magang di PT Ritzproject yakni:

Tabel 3.1 Waktu Saat Pelaksanaan Magang Perusahaan

Pekerjaan	Minggu	Tanggal Mulai	Tanggal Selesai
Fase Breafing Project			
Pemberian informasi awal mengenai proyek untuk melakukan analisis kebutuhan, instalasi perangkat lunak, dan penyusunan dokumen spesifikasi program (SP).	1	26 Juni 2023	3 Juli 2023
Pembuatan register <i>login form admin</i>	2	4 Juli 2023	6 Juli 2023
Setting github dan <i>wishlist</i>	2	7 Juli 2023	7 Juli 2023
Fase Development Admin & User (Back-End)			
Pengerjaan bagian <i>login admin</i>	3	10 Juli 2023	15 Juli 2023

Pengerjaan bagian pemisahan hak akses <i>user</i> dan <i>admin</i> melalui <i>login as</i>	4	17 Juli 2023	18 Juli 2023
Pengerjaan bagian sistem <i>login admin</i> dan perubahan relasi database <i>admin</i>	4	20 Juli 2023	22 Juli 2023
Pengerjaan bagian 2 <i>role admin</i> dan <i>user</i> dalam satu tabel <i>users</i> di database untuk <i>problem solving error login admin</i>	5	24 Juli 2023	29 Juli 2023
Fase Development Master Discount Admin (Back-End)			
Pengerjaan bagian <i>master controller, edit,</i> dan <i>delete master discount</i>	6	31 Juli 2023	5 Agustus 2023
Pengerjaan bagian memperbaiki <i>error</i> pada menu <i>discount blade</i> bagian <i>edit</i> dan <i>delete</i> serta menerapkan memvalidasi pada <i>master discount blade</i>	7	7 Agustus 2023	12 Agustus 2023
Pengerjaan bagian <i>validation percentage</i> dan <i>value</i> pada bagian <i>master discount blade</i>	8	14 Agustus 2023	19 Agustus 2023
Fase Development Master Operational Admin (Back-End)			
Pengerjaan bagian tabel menu <i>master operational</i>	9	21 Agustus 2023	26 Agustus 2023
Fase Development Master Sales Admin (Back-End)			
Pengerjaan bagian menu <i>master sales</i> di bagian <i>master controller</i> untuk penyesuaian dari database	10	28 Agustus 2023	2 September 2023
Pengerjaan bagian <i>master sales blade</i> untuk menampilkan tabel	11	4 September 2023	9 September 2023

Pengerjaan bagian <i>master sales</i> di bagian <i>action edit dan delete</i>	12	11 September 2023	16 September 2023
Pengerjaan bagian validasi dari menu <i>master sales</i> pada <i>insert data</i>	13	18 September 2023	23 September 2023
Pengerjaan bagian <i>delete</i> di menu <i>master sales</i> dan menambahkan <i>button back</i> pada semua menu master dibagian <i>edit dan delete</i>	14	25 September 2023	30 September 2023
Fase Bugs Fixing (Back-End)			
Pengerjaan bagian <i>insert data</i> di menu <i>master discount</i> untuk <i>disabled</i> dan <i>Bugs Fixing</i>	15	2 Oktober 2023	7 Oktober 2023
Pengerjaan bagian mengantikan warna <i>button back</i> orange menjadi warna biru dan <i>Bugs Fixing</i>	16	9 Oktober 2023	14 Oktober 2023
Merevisikan menu <i>master</i> bagian admin di <i>delete</i> , menambahkan <i>button back</i> yang belum ada, dan <i>bugs fixing</i>	17 - 18	16 Oktober 2023	27 Oktober 2023
<i>Phase Feedback UAT Result</i>	19 - 22	31 Oktober 2023	24 November 2023

1. 26 Juni – 3 Juli

Selama minggu pertama, penulis dengan tim dan PMO PT Ritzproject melaksanakan bagian *brainstorming* guna megembangkan sistem *e-commerce*. Orientasi dilaksanakan melalui platform Google Meet setiap harinya, dengan durasi sekitar dua jam, bertujuan untuk berbagi informasi dan saling belajar untuk memahami sistem perusahaan. Dengan bimbingan dari

supervisor berpengalaman, mahasiswa dan tim mendapatkan pemahaman yang lebih mendalam tentang alat bantu yang digunakan untuk mendukung proyek yang akan dijalankan.

Selama periode satu minggu orientasi, dilakukan pertemuan daring menggunakan Zoom setiap harinya dengan durasi sekitar 2-3 jam. Tujuan pertemuan tersebut adalah untuk meningkatkan pemahaman peserta magang terkait lingkungan kerja, memperkuat hubungan antar tim, dan memahami alat serta teknologi yang akan digunakan dalam menjalankan tugas yang diberikan.

Dalam proses instalasi perangkat lunak yang dibutuhkan *tools*, termasuk DBeaver, composer, XAMPP, Visual Studio Code, dan Github. Sebagai pengembang perangkat lunak di bagian belakang (*back-end*), penulis perlu menginstal semua perangkat lunak, tidak hanya alat-alat *back-end* saja. Hal ini disebabkan oleh fakta bahwa tahap desain antarmuka pengguna (UI/UX) untuk *front-end* dan tahap pengujian nantinya melibatkan seluruh sistem secara menyeluruh.



Gambar 3.2 Dbeaver Tools

Sumber: enterprisedb.com

Gambar 3.2 merupakan *tools* DBeaver yang diperlukan oleh *developer* yang mengelola dan mengakses database. *Tools* DBeaver dapat mendukung banyak jenis database dari mulai MySQL, Postgre, MS SQL Server, dan bahkan mendukung juga database nosql seperti mongodb. DBeaver juga

punya banyak versi yang dapat dijalankan di berbagai sistem operasi seperti mengakses DBeaver kepada aplikasi Visual Studio Code agar memudahkan dalam pembuatan aplikasi maupun website.



Gambar 3.3 Composer Tools

Sumber: getcomposer.org

Gambar 3.3 merupakan *tools* Composer yang diperlukan oleh *developer* untuk menginstalasikan pada *tools* Composer yang berfungsi untuk membuat para *developer* menggunakan Bahasa PHP menjadi hidup lebih sederhana. Dengan adanya composer sebagai *package-manager library*, para untuk *developer* dapat memudahkan membuat program yang sering dipakai dalam beberapa proyek seperti membuat website.



Gambar 3.4 Xampp Tools

Sumber: badoystudio.com

Gambar 3.4 merupakan *tools* Xampp yang diperlukan oleh *developer* untuk menginstalasikan pada *tools* Xampp yang berfungsi untuk server mandiri (localhost), sistem ini terdiri dari perangkat lunak Apache HTTP Server, basis data MySQL, dan interpreter bahasa yang *code* menggunakan seperti PHP dan Perl.



Gambar 3.5 Visual Studio Code Application

Sumber: techcelup.com

Gambar 3.5 merupakan aplikasi Visual Studio Code yang penting untuk diperlukan saat *developer* membuat website. Pada aplikasi Visual Studio Code yang berfungsi untuk melakukan pengerjaan proyek Visual Studio Code sebagai alat penyunting *code* yang mendukung berbagai bahasa

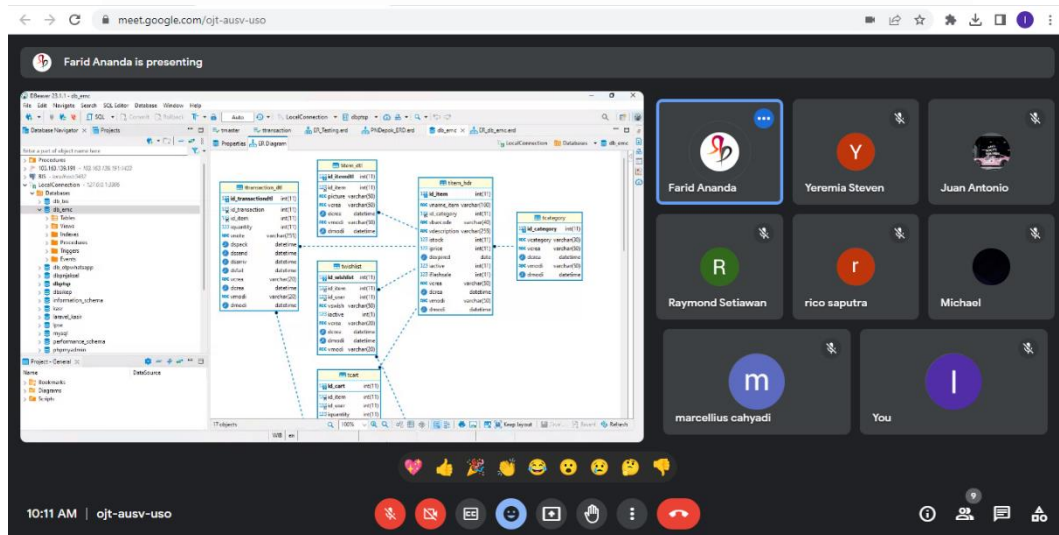
pemrograman. Aplikasi ini tidak hanya mampu membangun, tetapi juga memudahkan proses debugging aplikasi berbasis website.



Gambar 3.6 Github tools

Sumber: itbox.id

Gambar 3.6 merupakan *tools* Github yang diperlukan oleh setiap *developer* untuk mengumpulkan *task-task* agar mengabungkan pada mengerjakan fitur masing-masing.



Gambar 3.7 Meeting saat breafing perdana

Gambar 3.7 merupakan pertemuan awal untuk mendapatkan arahan bersama PMO PT Ritzproject melibatkan bagian *brainstorming*. Dalam

meeting nantinya akan memberitahukan kepada penulis dan tim untuk membahas yang cukup *detail* dari perancangan dan pengembangan aplikasi berbasis website sistem *e-commerce* yang akan dikerjakan selama periode magang ini. Tujuan *meeting* *breafing* ini mendapatkan pengalaman juga bersama PMO PT Ritzproject yang membekali untuk dunia perkerjaan setelah lulus, mendapatkan ilmu yang di jelaskan oleh PMO PT Ritzproject secara berpikir kritis bersama tim untuk perancangan kedepannya yang menuju yang baik serta memberikan arahan-arahan yang benar oleh PMO PT Ritzproject akan diberikan *task-task* oleh penulis dan tim melalui Google meet.

2. 4 Juli – 6 Juli

Pada minggu kedua *software developer* melakukan mendesign tampilan website *e-commerce* untuk mengembangkan dari *batch* sebelumnya. Pekerjaan yang dilakukan pada minggu kedua ini log in as untuk register admin.

```
<!-- 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 PARTNER!</p>
      </div>

      <div class="col-md-6">
        <div class="row">
          <p class="h2 fw-bold mt-3">Login Admin</p>
        </div>
        <form method="POST" action="{{ url('proses_login_admin') }}">
          @if (Session::get('info'))
            <div class="alert alert-info">
              {{ Session::get('info') }}
            </div>
          @endif
        </form>
      </div>
    </div>
  </div>
</section>
```

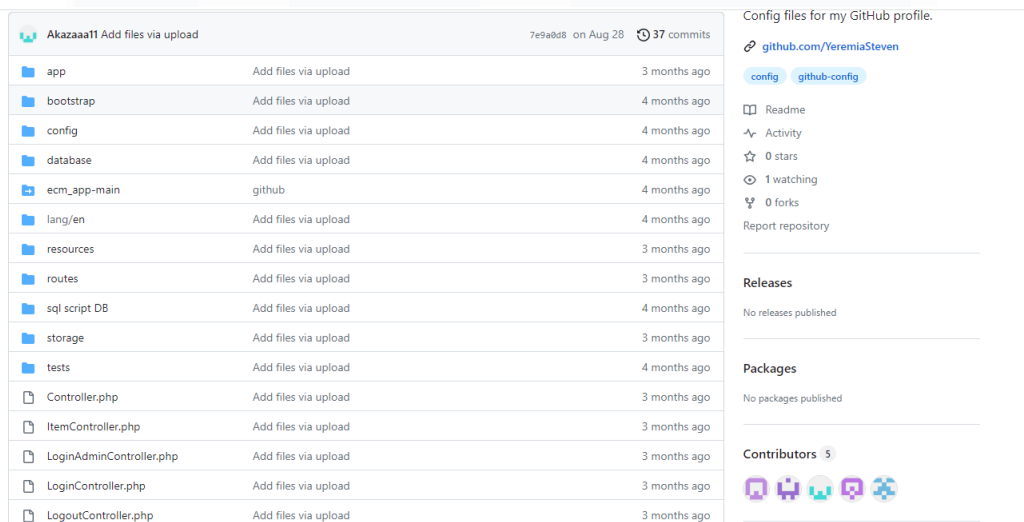
Gambar 3.8 Code HTML login register admin

Gambar 3.8 merupakan code HTML login form penempatan register untuk admin. Dalam code ini memiliki section yaitu login form mengelompokkan halaman yang terkait secara tematik atau fungsional, Background dapat menyesuaikan dengan tema, memiliki banyak class, max-width digunakan untuk mengatur besar kecilnya pada halaman, memiliki

action untuk mengatur link pada proses login admin, dan arlet digunakan untuk sebagai pop up apakah bisa dijalankan apa tidak. Pada login register admin yang digunakan untuk jika pengguna ingin login sebagai admin maka harus register terlebih dahulu.

3. 7 Juli

Pada minggu kedua *software developer* melakukan *setting* Github untuk mengupload file hasil progress pada saat setiap minggunya pada gambar 3.9.



Gambar 3.9 Website Github untuk mengupload progress file

Setelah melakukan setting Github *system analyst* dan *software developer* melakukan *wishlist* bagian user. *wishlist* pada website *e-commerce* yang membuat developer dan tim. Dokumen Spesifikasi Program yang disusun oleh *System analyst* berfungsi sebagai panduan bagi pengembang saat melakukan pengembangan sistem *e-commerce*.

```

@extends('template_user')
@section('content')

<!-- Products -->
<section id="products" style="background-color:white; padding-top: 2rem;
">
  <div class="container-fluid">
    <div class="row mt-5">
      <div class="col-md d-flex justify-content-center">
        <p class="h3 fw-bold me-3">Wishlist <i class="fa-solid fa-heart" style="color: #ff0000;"></i></p>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md">
      <div
        id="carouselExampleSlidesOnly"
        class="carousel slide"
        data-bs-ride="carousel"
      >
        <div class="carousel-inner">
          <div class="carousel-item active">
            @if (empty($item))
              <div class="row mt-3 mb-5 justify-content-center bg-light rounded-3">
                <p class="fs-5 text-center fw-bolder my-5">Wishlist Kamu Kosong</p>
              </div>
            @endif
          </div>
        </div>
      </div>
    </div>
  </div>

```

Gambar 3.10 Code wishlist.blade pada produk

Gambar 3.10 merupakan code wishlist.blade pada produk yang digunakan untuk menyimpan produk-produk yang disukai pengguna. Dalam code ini memiliki template_user yang digunakan untuk menampilkan halaman di bagian user, Memiliki section yakni products pada database, background dapat menyesuaikan tema, memiliki banyak class, untuk color pada button wishlist berwarna merah jika di klik, dan carousel digunakan untuk menampilkan gambar sebuah produk.

4. 10 Juli – 15 Juli

Pada minggu ketiga *software developer* melakukan pengerjaan bagian login admin. Proyek ini memanfaatkan HTML, CSS, dan JavaScript, dengan tambahan penggunaan bootstrap. Hasil akhir dari sistem *e-commerce* tidak akan sepenuhnya mirip dengan tampilan yang membuat di halaman *front-end* perlu diubah sesuai dengan persetujuan *System analyst* dan disesuaikan dengan kerangka kerja Laravel oleh pengembang *back-end*. Selama rentang waktu 5 hari, penulis fokus pada pengembangan *back-end* untuk halaman

login untuk admin. Halaman-halaman ini berlaku untuk semua peran dan akan ada pemisahan antara login admin.

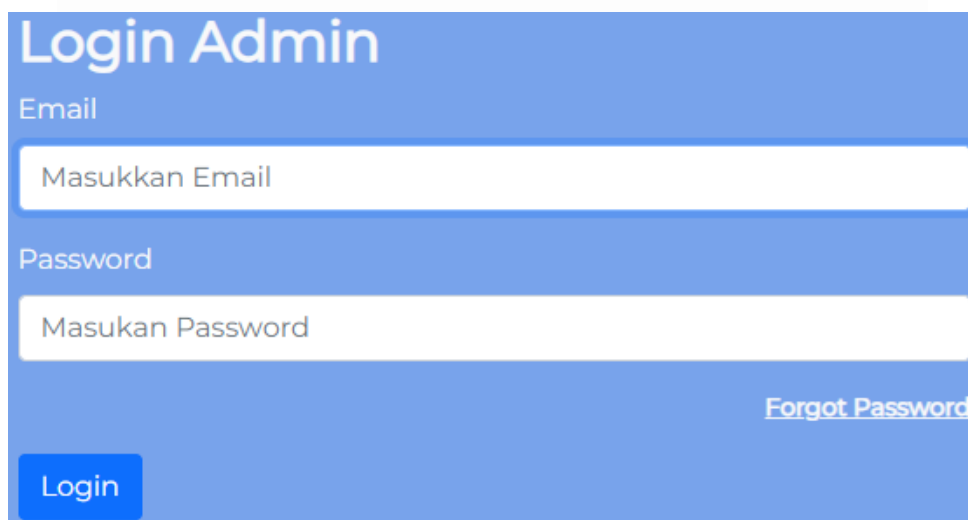
```
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Email</label>
  <input id="email" type="text" class="form-control" name="email" value="{{ old('email') }}" placeholder="Masukkan Email" required autocomplete="current-password">
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Password</label>
  <input id="password" type="password" class="form-control" name="password" placeholder="Masukan Password" required autocomplete="current-password">
</div>
<div class="row mt-3">
  <p class="small text-end"><a href="forgot_password" class="text-light fw-bold" >Forgot Password</a></p>
</div>

<button type="submit" class="btn btn-primary btn-user btn-block">
  Login
</button>
</form>
<div class="row mt-3">
  <p class="small text-center">Don't have an account yet? <a href="register" class="text-light fw-bold" >Register</a></p>
</div>
</div>
```

Gambar 3.11 Code Halaman login admin

Gambar 3.11 merupakan bagian code halaman login form admin untuk judul form yang dapat mengenali login sebagai admin. Navbar untuk tata letak bagian login as untuk nama, yang tertanam di salah satu bagian layar website akan menentukan sistem navigasi yang dikumpulkan, memiliki banyak class, form-label untuk menamakan pada textbox, required autocomplete untuk jika berhasil memasukan email dan password saat login admin, ada row pada forgot password, dan button login.

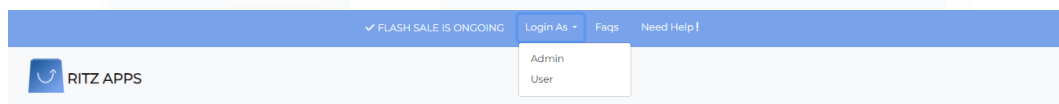


Gambar 3.12 Code tampilan login admin

Gambar 3.12 merupakan tampilan pada code login admin yang menampilkan pada router HTML. Login admin digunakan untuk saat pengguna ingin login sebagai admin.

5. 17 Juli – 18 Juli

Pada minggu keempat *software developer* melakukan pengerjaan bagian pemisahan hak akses user dan admin melalui login as. Selama 2 hari *back-end* melakukan pengembangan login user dan admin harus memisahkan melalui login as agar dapat memudahkan pengguna untuk login.



Gambar 3.13 Code tampilan login as admin dan user

Gambar 3.13 merupakan tampilan login as *admin* dan *user* yang dapat memisahkan untuk login agar pengguna dapat mudah membedakan login *user* dan *admin*.

```
<!-- 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 me-1"></i>FLASH SALE IS ONGOING</a>
      <div class="btn-group">
        <button class="btn btn-sm dropdown-toggle text-light small" type="button" data-bs-toggle="dropdown" aria-expanded="false">
          Login As
        </button>
        <ul class="dropdown-menu">
          <li><a class="dropdown-item small" href="loginadmin">Admin</a></li>
          <li><a class="dropdown-item small" href="login">User</a></li>
        </ul>
      </div>
    </div>
  </nav>
</section>
```

Gambar 3.14 Code login as admin dan user

Gambar 3.14 merupakan bagian tampilan dan code pemisahan login *admin* dan *user*. Code login as terdiri atas 1 *section*, yakni navbar. Navbar untuk tata letak bagian login as untuk nama, yang tertanam di salah satu bagian layar website akan menentukan sistem navigasi yang dikumpulkan, memiliki banyak class, button login as untuk memilih login admin atau user melalui combobox, dan href menampilkan link login pada router. Code login as admin dan user ada halaman dropdown menu yang digunakan untuk dapat

mempelihatkan menu seperti pemilihan login as ingin melalui login *user* atau login *admin*.

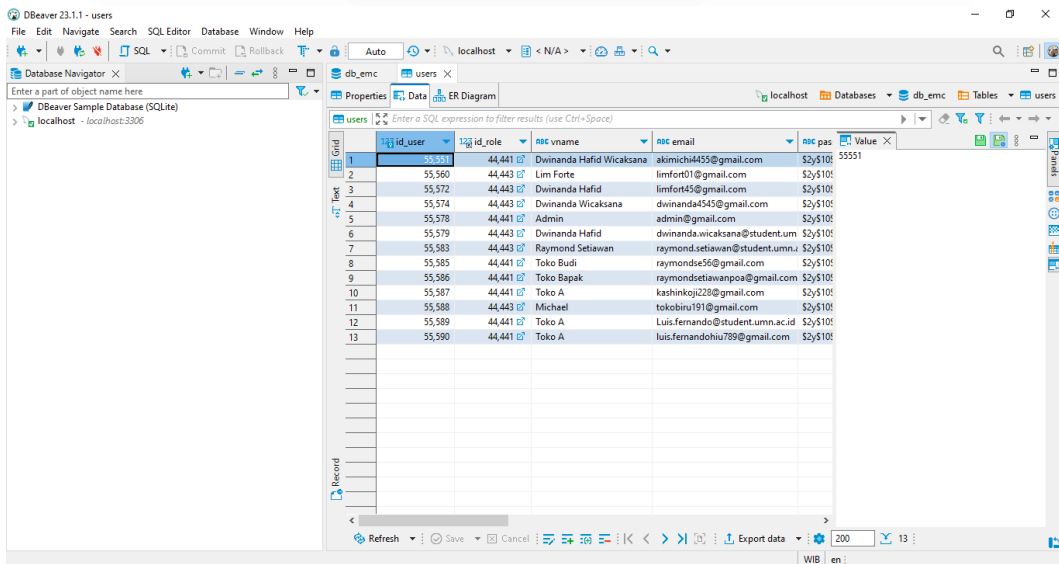
```
//Route Admin
Route::get('loginadmin', $ctrl->\LoginAdminController@index')
-> middleware('guest')
-> name('loginadmin');
```

Gambar 3.15 Code router admin

Gambar 3.15 merupakan code href ini dapat digunakan untuk melihat halaman link pada router yang dapat memisahkan dalam halaman yang berbeda pada login admin.

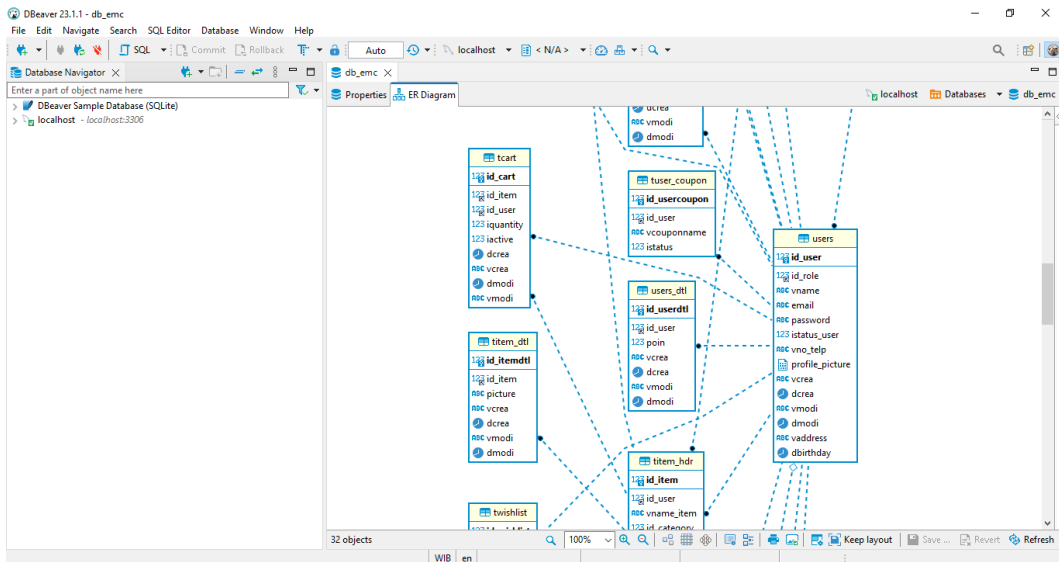
6. 20 Juli – 22 Juli

Pada minggu keempat *software developer* melakukan pengerjaan bagian sistem login admin dan perubahan relasi database admin. Selama 3 hari *system analyst* bersama *software developer* melakukan perubahan relasi database bagian admin untuk membedakan untuk login user dengan admin.



Gambar 3.16 Sistem role admin dan user

Gambar 3.16 merupakan sistem data admin pada tabel user untuk memisahkan *user dan admin* dengan adanya *id_role*.



Gambar 3.17 Relasi database admin

Gambar 3.17 merupakan relasi database admin yang masuk kedalam tabel user untuk mengakses keseluruhan data-data pada website *e-commerce*.

7. 24 Juli – 29 Juli

Pada minggu kelima *software developer* melakukan pengerjaan bagian *2 role admin dan user* dalam satu tabel *users* di database untuk *problem solving error login admin*.

id_user	id_role	vname	email	password
55,551	44,443	Dwinanda Hafid Wicaksana	akimichi4455@gmail.com	\$2y\$10\$
55,560	44,443	Lim Forte	limfort01@gmail.com	\$2y\$10\$
55,572	44,443	Dwinanda Hafid	limfort45@gmail.com	\$2y\$10\$
55,574	44,443	Dwinanda Wicaksana	dwinanda4545@gmail.com	\$2y\$10\$
55,578	44,441	Admin	admin@gmail.com	\$2y\$10\$
55,579	44,443	Dwinanda Hafid	dwinanda.wicaksana@student.um	\$2y\$10\$
55,583	44,443	Raymond Setiawan	raymond.setiawan@student.um	\$2y\$10\$
55,585	44,441	Toko Budi	raymondse56@gmail.com	\$2y\$10\$
55,586	44,441	Toko Bapak	raymondsetiawanpoa@gmail.com	\$2y\$10\$
55,587	44,441	Toko A	hashinkoj22@gmail.com	\$2y\$10\$
55,588	44,443	Michael	tokobini191@gmail.com	\$2y\$10\$
55,589	44,441	Toko A	Luis.fernando@student.um.ac.id	\$2y\$10\$
55,590	44,441	Toko A	luis.fernandohiu789@gmail.com	\$2y\$10\$

Gambar 3.18 Data tabel user pada id_role

Gambar 3.18 merupakan pembagian 2 *role admin* dan *user* dalam 1 tabel *users* yang dapat membedakan *id_role* yakni 44441 untuk *admin* serta 44443 untuk *user* agar dapat menjadi solusi *error login admin*.

8. 31 Juli – 5 Agustus

Pada minggu keenam *software developer* melakukan pengerjaan bagian *master controller, edit, dan delete master discount*.

```
//MasterDiskon
public function index_discount(){
    $nama_web = DB::table('tglobalsetting')->where('id_global',3)->first();
    $master = DB::table('tdiscount')
    //->join('ttransaction_event' , 'tdiscount.id_discount', '=', 'ttransaction_event.id_discount')
    //->get(['tdiscount.*', 'ttransaction_event.id_discount']);
    ->get();

    return view('masterdiscount',compact ('master','nama_web'));
}

public function store_discount(Request $request){
    $query = DB::table('tdiscount')
    ->insert([
        'vdesc' => $request->description,
        'percentage' => $request->percentage,
        'value' => $request->value,
        'vcrea' => Auth::user()->email,
        'dcrea' => Carbon::now()
    ]);
    if (!$query){
        Alert::error('Error', 'Data cannot be store');
    } else {
        Alert::Success('success', 'Data has been store');
    }
    return redirect('master/discount');
}
```

Gambar 3.19 Code *mastercontroller* bagian *master discount*

Gambar 3.19 merupakan code *mastercontroller* bagian *master discount* ada halaman berupa *public function* *index* yang digunakan untuk menjalankan peran-peran tertentu, dengan parameter *\$request* dari fungsi tersebut. Parameter *\$request* kemudian diatur ke nilai setiap nama properti yang disimpan dalam parameter *\$request*. Setiap atribut divalidasi dengan pengecekan *query* untuk memasukan database melalui *join* dan *get*, dan arlet untuk mengetahui apakah berhasil dijalankan apa tidak. kemudian *\$request* dapat memeriksa apakah inputan tersebut terdapat dalam kumpulan data tabel

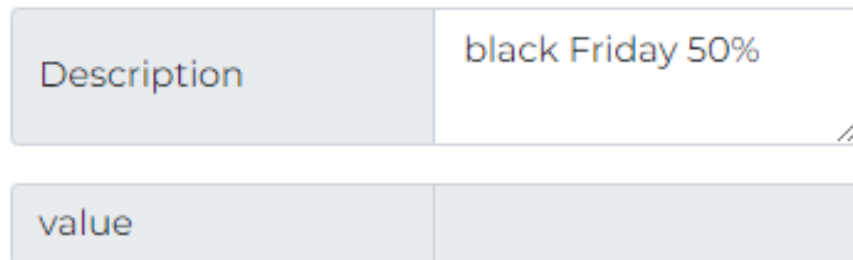
master diskon pada database. *Controller* ini berfungsi untuk pengontrol alur aplikasi antara *router* dan *view*. *Controller* dapat mengatur permintaan dari pengguna sehingga dapat merespon *router* dan *view*. Insert digunakan untuk memasukan dari data database yang disesuaikan dengan fitur master diskon. Return redirect yang digunakan untuk menampilkan halaman link pada master diskon.

```
</div>
<div class="input-group mb-3 mt-3">
  <span class="input-group-text w-50" id="inputGroup-sizing-default"
    >Description</span
  >
  <textarea
    name="description"
    type="text"
    class="form-control"
    aria-label="Sizing example input"
    maxlength="255"
    aria-describedby="inputGroup-sizing-default"required
  >{{ $master->vdesc}}</textarea>
</div>
<div class="input-group mb-3 mt-3">
  <span class="input-group-text w-50" id="inputGroup-sizing-default"
    >value</span
  >
  <input
    name="value"
    type="number"
    min="1"
    class="form-control"
    value="{{ $master->value }}"
    aria-label="Sizing example input"
    aria-describedby="inputGroup-sizing-default"
    required
  />
</div>
```

Gambar 3.20 Code view bagian masterdiscount_editblade

Gambar 3.20 merupakan code *view* bagian master discount_editblade bagian description dan value untuk menampilkan sebuah form untuk mengedit diskon yang ditentukan. Code masterdiscount_editblade ada halaman berupa memiliki banyak class, Span yang digunakan untuk memberikan pembagian pada judul form yang ditampilkan. Input yang

digunakan sebagai mengatur dalam berbagai code dapat diatur agar data bisa menyesuaikan dengan form yang ditentukan, dan type untuk memilih tipe angka maupun huruf.



The image shows two input fields. The top field is labeled 'Description' and contains the text 'black Friday 50%'. The bottom field is labeled 'value' and is currently empty.

Gambar 3.21 Code tampilan description dan value

Gambar 3.21 merupakan code tampilan description dan value pada master discount edit dapat melihat lagi setelah mengisi insert data dari tabel master discount agar jika ada kesalahan menginput dapat mengeditnya. Dalam tampilan master discount edit yang developer kerjakan hanya pada code tampilan description untuk membuat event dan value membuat diskon yang berbentuk rupiah.

```
<!-- Menghubungkan dengan view template master -->
@extends('template')
@section('content')
<div class="card-body">
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-4 ">
        <form action="{{ url('master/discount/delete')}}" method="post" enctype="multipart/form-data">
          @csrf
          <div class="form-group">
            <label class="me-3" for="">Are you sure delete this data?</label>
          </div>
          <div class="form-group">
            <button type="submit" class="btn btn-warning">Delete</button>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
</div>
@endsection
```

Gambar 3.22 Code view bagian masterdiscount_deleteblade

Gambar 3.22 merupakan code *view* bagian masterdiscount_deleteblade untuk menampilkan apakah ingin menghapus kolom tabel diskon. Code masterdiscount_deleteblade ada halaman berupa *template*, *form action*, dan

hidden dapat menghapus kolom data tabel diskon. *Template* pada *extends* yang dapat berfungsi untuk menampilkan tampilan halaman baru. *Form action* digunakan untuk menyimpan link pada router yang dapat membuat halaman link baru sesuai code yang telah dibuat. Dan *hidden* digunakan saat telah menekan tombol delete maka tampilan kolom akan di hapus database akan terotomatis hapus juga.

9. 7 Agustus – 12 Agustus 2023

Pada minggu ketujuh *software developer* melakukan pengerjaan bagian memperbaiki *error* pada menu discount blade bagian edit dan delete serta menerapkan menvalidasi pada master discount blade.

```
<div class="col d-flex flex-column h-sm-100">
  <h1 class="text-center mt-3 fw-bold">Master Discount</h1>
  <div class="col-md-4 mt-3 ps-5">
    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#add_discount">Insert Data</button>
  </div>
  <div class="col-md ps-5 pe-5" style="padding-top:1rem;">
    <table id="example" class="table table-striped" >
      <thead>
        <tr class="text-center">
          <th style="width:70%">id Discount</th>
          <th style="width:75%">Discount Percentage</th>
          <th style="width:200%">Description</th>
          <th style="width:70%">Price</th>
          <th style="width:70%">Created by</th>
          <th style="width:100%">Created Date</th>
          <th style="width:70%">Modified by</th>
          <th style="width:100%">Last Update</th>
          <th style="width:50%">Action</th>
        </tr>
      </thead>
      <tbody>
        @foreach($master as $i => $u)
          <tr>
            <td>{{++$i}}</td>
            <td>{{ $u->percentage}}</td>
            <td>{{ $u->vdesc}}</td>
            <td>{{ $u->value}}</td>
            <td>{{ $u->vcrea}}</td>
          </tr>
        </tbody>
      </table>
    </div>
</div>
```

Gambar 3.23 Code Masterdiscountblade

Gambar 3.23 merupakan code masterdiscountblade untuk menampilkan pengisian form melalui *insert data* dan dapat menampilkan tabel setelah mengisi *insert data*. Coding masterdiscountblade ada halaman berupa *script*, *href*, dan *modal* yang untuk menampilkan master discount. Dalam membuat tampilan halaman tabel ini struktur thead dan tbody disusun dengan mengandalkan elemen-elemen HTML seperti tr, th, dan td, yang ditambahkan dengan class-class dari bootstrap untuk mempermudah

penerapan CSS dan JavaScript. *Script* yang dapat digunakan memvalidasikan data jika pada saat tidak mengisi pada form yang ditampilkan maka ada sebuah munculnya tulisan yang wajib diisi oleh pengguna tersebut.

```
//Master Discount
Route::get('master/discount', 'App\Http\Controllers\MasterController@index_discount');
Route::post('master/discount/store', 'App\Http\Controllers\MasterController@store_discount');
Route::get('master/discount/edit/{id}', 'App\Http\Controllers\MasterController@index_discountup');
Route::post('master/discount/edit', 'App\Http\Controllers\MasterController@update_discount');
Route::get('master/discount/delete/{id}', 'App\Http\Controllers\MasterController@index_discountdt');
Route::post('master/discount/delete', 'App\Http\Controllers\MasterController@delete_discount');
```

Gambar 3.24 Code router master discount di web.php

Gambar 3.24 merupakan code *Href* yang digunakan untuk menampilkan sebuah halaman baru bagi master discount menggunakan *router* melalui web.php yang berupa link halaman baru yaitu index pada tampilan tabel, index pada tampilan edit, dan index pada tampilan delete.

```
<!-- Modal -->
<div class="modal fade" id="add_discount" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Insert Data Discount</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form method="POST" action="{{ url('master/discount/store')}}">
          @csrf
        </form>
      </div>
    </div>
  </div>
</div>
```

Gambar 3.25 Code modal untuk pop up insert data dibagian masterdiscount.blade

Gambar 3.25 merupakan code *modal* yang digunakan untuk memberikan sebuah *pop up* pada *insert data* dibagian masterdiscount.blade. Berfungsi untuk menampilkan sebuah *form* untuk mengisi *form* untuk tabel master discount pada *admin* yang dapat memberikan diskon event kepada *user* sebagai pembeli, dan action untuk menyimpan link index pada *router*.

id Discount	Discount Percentage	Description	Price	Created by	Created Date	Modified by	Last Update	Action
1	5.9	Potongan 5.9			16-10-2023			 
2		Potongan 1000	1000		16-10-2023			 
3	10	Potongan 10%			16-10-2023			 
4	20	Potongan 20%			16-10-2023			 
5		Potongan 3000	3000		16-10-2023			 
6	50	black Friday 50%		luis.fernando dohiu789 @gmail.com	23-11-2023			 

Showing 1 to 6 of 6 entries

Previous **1** Next

Gambar 3.26 Code tampilan master discount

Gambar 3.26 merupakan code tampilan menu master discount dalam perbaikan *error edit* dan *delete* dibagian *masterdiscountblade* karena *error* tidak ada tampilan *action*, fungsi *action* ini dapat mengubah dan menghapus pada isi tabel master diskon.

Insert Data Discount
✕

Discount Percentage

Description

Value

Close
Save

Gambar 3.27 Code tampilan pop up insert data untuk validasi

Gambar 3.27 merupakan tampilan pop up insert data ini menerapkan memvalidasi pengisian insert data berupa discount percentage, description, dan value untuk saat tidak di isi maka ada tulisan untuk mengisi pada master diskon.

10. 14 Agustus – 19 Agustus 2023

Pada minggu kedelapan *software developer* melakukan pengerjaan bagian validation percentage dan value pada bagian master discount blade.

```
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Discount Percentage</label>
  <input name="percentage" type="number" inputmode="numeric" min="1" pattern="[0-9]*" max="100" class="form-control" placeholder="Enter the Di
</div>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Description</label>
  <textarea name="description" type="text" rows="1" class="form-control" placeholder="Enter your Description" maxLength = "255" required>
</div>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Value</label>
  <input name="value" type="number" class="form-control" min="1" placeholder="Enter the Value" required>
</div>
```

Gambar 3.28 Code masterdiscountblade untuk validasikan bagian percentage dan value

Gambar 3.28 merupakan code masterdiscountblade untuk memvalidasikan bagian percentage dan value. Code masterdiscountblade ada halaman berupa name, type, inputmode, min, max, pattern, dan placeholder dengan menyesuaikan pengisian untuk form insert data yang dapat menginputkan data diskon sesuai event pada masterdiskonblade.

11. 21 Agustus – 26 Agustus 2023

Pada minggu kesembilan *software developer* melakukan pengerjaan bagian tabel menu *master operational*.

```

<!-- Menghubungkan dengan view template master -->
@extends('template')
@section('content')

<div class="col d-flex flex-column h-sm-100">
  <h1 class="text-center mt-3 fw-bold">Master Operational Hour</h1>
  <div class="col-md-4 mt-3 ps-5">
    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#add_operational">Insert Data</button>
  </div>
  <div class="col-md ps-5 pe-5" style="padding-top:1rem;">
    <table id="example" class="table table-striped" style="width:100%">
      <thead>
        <tr class="text-center">
          <th scope="col">Id Operational</th>
          <th scope="col">Name User</th>
          <th scope="col">Date</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">Action</th>
        </tr>
      </thead>
      <tbody>
        @foreach($master as $i => $u)
          <tr>
            <td>{{++$i}}</td>
            <td>{{ $u->vname}}</td>
            <td>{{ date('d-m-Y', strtotime($u->doperational)) }}</td>
            <td>{{ $u->time_open}}</td>
            <td>{{ $u->time_close}}</td>
            <td>{{ $u->vmodi}}</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>

```

Gambar 3.29 Code masteroperationalblade bagian tabel master operational

Gambar 3.29 merupakan code masteroperationalblade bagian tabel master operational yang dapat menampilkan tabel pada jam operasional toko. Code masteroperationalblade ada halaman berupa template dan href dari membuat halaman tabel ini dapat struktur dari thead dan tbody, lalu di dalamnya terdapat tr, th, dan td yang sesuai HTML. *Template* yang dapat berfungsi untuk menampilkan tampilan halaman baru.

```

//Master Operational Hour
Route::get('master/operational', 'App\Http\Controllers\MasterController@index_operational');
Route::post('master/operational/store', 'App\Http\Controllers\MasterController@store_operational');
Route::get('master/operational/edit/{id}', 'App\Http\Controllers\MasterController@index_operationalupdate');
Route::post('master/operational/edit', 'App\Http\Controllers\MasterController@update_operational');
Route::get('master/operational/delete/{id}', 'App\Http\Controllers\MasterController@index_operationaldelete');
Route::post('master/operational/delete', 'App\Http\Controllers\MasterController@delete_operational');

```

Gambar 3.30 Code router operational hour dibagian web.php

Gambar 3.30 merupakan code router operational hour dibagian web.php pada link href master operational hour yang digunakan untuk membuat link sesuai dengan template tabel master operational hour. Router operational hour berupa link halaman baru yaitu index pada tampilan tabel, index pada tampilan edit, dan index pada tampilan delete.

Id Operational	Name User	Date	Time Open	Time Close	Modified By	Last Update	Action
1	Toko Bapak	10-10-2023	13:01:00	19:01:00	raymondsetiawanpoa@gmail.com	09-10-2023	
2	Toko A	02-08-2023	05:00:00	17:00:00			
3	Toko A	03-08-2023	05:00:00	17:00:00			
4	Toko A	04-08-2023	05:00:00	17:00:00			
5	Toko A	05-08-2023	05:00:00	17:00:00			
6	Toko A	06-08-2023	05:00:00	17:00:00			
7	Toko A	07-08-2023	05:00:00	17:00:00			
8	Toko A	08-08-2023	05:00:00	17:00:00			
9	Toko A	09-08-2023	05:00:00	17:00:00			
10	Toko A	10-08-2023	05:00:00	17:00:00			

Showing 1 to 10 of 33 entries

Previous **1** 2 3 4 Next

Gambar 3.31 Code tampilan tabel master operational hour

Gambar 3.31 merupakan code tampilan tabel master operational hour dapat menyesuaikan jam toko buka sampai toko tutup pada login admin. Dalam mengatur jam operational bagian admin dapat menampilkan jam operational untuk user sehingga user dapat melihat jam pada toko-toko tersebut.

Buku

Stock : 989

~~Rp. 2,000~~ Rp. 1,880

Toko Bapak

11:00 - 17:00

Wishlist

Description

Test1

Gambar 3.32 Code tampilan operational hour toko pada user

Gambar 3.32 merupakan code tampilan operational hour toko pada user yang menampilkan berbagai toko yang dapat menjadwalkan saat toko buka sampai tutup toko.

12. 28 Agustus – 2 September 2023

Pada minggu kesepuluh *software developer* melakukan pengerjaan bagian menu master sales di bagian master controller untuk penyesuaian dari database.

```
//Master Sales
public function index_sales(){
    $nama_web = DB::table('tglobalsetting')->where('id_global',3)->first();
    $master = DB::table('tsales')
        ->join('users', 'tsales.id_user', '=', 'users.id_user')
        ->get(['tsales.*', 'users.vname']);
    return view('mastersales',compact ('master','nama_web'));
}

public function insert_sales(Request $request) {
    $query = DB::table('tsales')
    ->insert([
        'id_user' => $request->user,
        'vnamesales' => $request->namesales,
        'vhp_sales' => $request->nosales,
        'isales_target' => $request->salestarget,
        'vcrea' => Auth::user()->email,
        'dcrea' => Carbon::now()
    ]);
    if (!$query){
        Alert::error('Error', 'Data cannot Insert');
    } else {
        Alert::Success('success', 'Data has been Insert');
    }
    return redirect('master/sales');
}
```

Gambar 3.33 Code mastercontroller bagian master sales

Gambar 3.33 merupakan code mastercontroller bagian master sales untuk memasukkan database yang menyesuaikan tabel, insert data, edit, dan delete. Code mastercontroller bagian master sales ada halaman berupa *public function* yang digunakan untuk menjalankan peran-peran tertentu, dengan parameter \$request dari fungsi tersebut. Parameter \$request kemudian diatur ke nilai setiap nama properti yang disimpan dalam parameter \$request. Setiap atribut divalidasi dengan pengecekan *query* untuk memasukan database melalui *join* dan *get*, dan arlet untuk mengetahui apakah berhasil dijalankan

apa tidak. kemudian `$request` dapat memeriksa apakah inputan tersebut terdapat dalam kumpulan data tabel master sales pada database. *Controller* ini berfungsi untuk pengontrol alur aplikasi antara router dan *view*. *Controller* dapat mengatur permintaan dari pengguna sehingga dapat merespon *router* dan *view*. Return redirect yang digunakan untuk menampilkan halaman link pada master diskon. Index ini dapat menampilkan secara baik dihalaman *browser* sesuai dengan router pada `web.php`

```
public function index_updatesales($id){
    $nama_web = DB::table('tglobalsetting')->where('id_global',3)->first();
    $master = DB::table('tsales')->where('id_sales',$id)->first();
    return view('mastersales_edit',compact('master','nama_web'));
}

public function update_sales(Request $request){
    $check = DB::table('tsales')->where(['id_sales'=>$request->id])->first();
    if (!$check){
        Alert::error('Error', 'Data cannot be change');
    } else{
        DB::table('tsales')->where(['id_sales'=>$request->id])
        ->update([
            'id_user' => $request->user,
            'vnamesales' => $request->namesales,
            'vhp_sales' => $request->nosales,
            'isales_target' => $request->salestarget,
            'vmodi' => Auth::user()->email,
            'dmodi' => Carbon::now()
        ]);
        Alert::Success('success', 'Data has been Updated');
    }
    return redirect('master/sales');
}
```

Gambar 3.34 Code syntax index update sales

Gambar 3.34 merupakan code syntax index update sales untuk memasukkan database yang menyesuaikan. Code syntax index update sales ada halaman berupa index untuk menampilkan sebuah halaman baru, tabel `tsales` dari database, *public function* yang digunakan untuk menjalankan peran-peran tertentu, dengan parameter `$request` dari fungsi tersebut. Parameter `$request` kemudian diatur ke nilai setiap nama properti yang

disimpan dalam parameter \$request. Setiap atribut divalidasi dengan pengecekan *query*, dan return redirect untuk halaman link pada router.

```
public function index_deletesales($id){
$nama_web = DB::table('tglobalsetting')->where('id_global',3)->first();
return view('mastersales_delete',compact('id','nama_web'));
}

public function delete_sales(Request $request){
$query = DB::table('tsales')->where('id_sales',$request->id)->delete();
$query = DB::table('tsales')->where('vnamesales',$request->id)->delete();
Alert::Success('success', 'Data has been deleted');

return redirect('master/sales');
}
}
```

Gambar 3.35 Code syntax index delete sales

Gambar 3.35 merupakan code syntax index delete sales sales untuk memasukkan database yang menyesuaikan. Code syntax index delete sales ada halaman berupa index untuk menampilkan sebuah halaman baru, tabel tsales dari database, *public function* yang digunakan untuk menjalankan peran-peran tertentu, dengan parameter \$request dari fungsi tersebut. Parameter \$request kemudian diatur ke nilai setiap nama properti yang disimpan dalam parameter \$request. Setiap atribut divalidasi dengan pengecekan *query*, dan return redirect untuk halaman link pada router.

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
123 id_sales	1	int(11)	[v]	[v]	PRI		auto_increment
123 id_user	2	int(11)	[v]	[]	MUL		
ABC vnamesales	3	varchar(100)	[v]	[]			
123 isales_target	4	int(11)	[v]	[]			
ABC category	5	varchar(100)	[]	[]		NULL	
ABC vhp_sales	6	varchar(12)	[v]	[]			
ABC vcrea	7	varchar(50)	[v]	[]			
dcrea	8	datetime	[v]	[]			
ABC vmodi	9	varchar(50)	[]	[]		NULL	
dmodi	10	datetime	[]	[]		NULL	

Gambar 3.36 Database tsales pada master sales

Gambar 3.36 merupakan database tsales pada master sales ini memiliki *column name*, *data type*, dan *not null* untuk menyesuaikan pada tabel, insert data, edit, dan delete agar dapat memudahkan saat menginputkan insert data pada tabel master sales.

13. 4 September – 9 September 2023

Pada minggu kesebelas *software developer* melakukan pengerjaan bagian master sales blade untuk menampilkan tabel.

```
<!-- Menghubungkan dengan view template master -->
@extends('template')
@section('content')

<div class="col d-flex flex-column h-sm-100">
  <main class="row overflow-auto flex-column">
    <h1 class="text-center mt-3 fw-bold">Sales</h1>
    <div class="col-md-4 mt-3 ps-5">
      <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#admin_store">Insert Data</button>
    </div>
    <div class="col-md ps-5 pe-5" style="padding-top:1rem;">
      <table id="example" class="table table-striped" style="width:100%">
        <thead>
          <tr class="text-center">
            <th scope="col">Id Sales</th>
            <th scope="col">Id User</th>
            <th scope="col">Name Sales</th>
            <th scope="col">Phone Number</th>
            <th scope="col">Sales Target</th>
            <th scope="col">Modif Date</th>
            <th scope="col">Modif By</th>
            <th scope="col">Action</th>
          </tr>
        </thead>
        <tbody>
          @foreach($master as $i => $u)
            <tr>
              <td>{{ $u->id_sales }}</td>
              <td>{{ $u->id_user }}</td>
              <td>{{ $u->vnamesales }}</td>
              <td>{{ $u->vhpsales }}</td>
              <td>{{ $u->isales_target }}</td>
            </tr>
          @endforeach
        </tbody>
      </table>
    </div>
  </main>
</div>
```

Gambar 3.37 Code mastersalesblade

Gambar 3.37 merupakan code mastersalesblade bagian tabel master sales untuk menampilkan tabel sales pada *admin* toko. Code mastersalesblade ada halaman berupa *template*, *href*, dan *modal* dari membuat halaman tabel ini dapat membuat tampilan halaman tabel ini struktur *thead* dan *tbody* disusun dengan mengandalkan elemen-elemen HTML seperti *tr*, *th*, dan *td*, yang ditambahkan. *Template* yang dapat berfungsi untuk menampilkan halaman baru.

```

<script>
document.addEventListener('DOMContentLoaded', function () {
    var namesalesInput = document.querySelector('#namesales');
    var nosalesInput = document.querySelector('#nosales');

    namesalesInput.addEventListener('input', function () {
        var maxLength = 100;
        if (this.value.length > maxLength) {
            this.value = this.value.substring(0, maxLength);
        }
    });

    nosalesInput.addEventListener('input', function () {
        var maxLength = 12;
        if (this.value.length > maxLength) {
            this.value = this.value.substring(0, maxLength);
        }
    });
});
</script>
</div>
</div>
@endsection

```

Gambar 3.38 Code script di mastersales.blade

Gambar 3.38 merupakan code script di mastersales.blade yang digunakan untuk memvalidasikan *form* pada insert data master sales telah mengatur namesales maksimal 100 huruf dan nosales 12 angka, *public function* yang digunakan untuk menjalankan peran-peran tertentu, dengan parameter.

```

//Master Sales
Route::get('master/sales', 'App\Http\Controllers\MasterController@index_sales');
Route::post('master/sales/store', 'App\Http\Controllers\MasterController@insert_sales');
Route::get('master/sales/delete/{id}', 'App\Http\Controllers\MasterController@index_deletesales');
Route::post('master/sales/delete', 'App\Http\Controllers\MasterController@delete_sales');
Route::get('master/sales/edit/{id}', 'App\Http\Controllers\MasterController@index_updatesales');
Route::post('master/sales/edit', 'App\Http\Controllers\MasterController@update_sales');

```

Gambar 3.39 Code router master sales di web.php

Gambar 3.39 merupakan code *router* master sales dibagian web.php pada link *href* master sales yang digunakan untuk membuat link sesuai dengan template tabel master sales. Router master sales berupa link halaman

baru yaitu index pada tampilan tabel, index pada tampilan edit, dan index pada tampilan delete.

```

<!-- Modal -->
<div class="modal fade" id="admin_store" tabindex="-1" aria-labelledby="exampleModallabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModallabel">Insert Data Sales</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form action="{{ url('master/sales/store')}}" method="POST">
          @csrf
          <div class="form-group row mt-3">
            <div class="mb-3">
              <label for="exampleFormControlInput1" class="form-label">Id User</label>
              <select name="user" class="form-select form-select-sm" aria-label=".form-select-lg example" required>
                <option value="">Choice your Id User</option>
                <option value="55560">55560</option>
                <option value="55572">55572</option>
                <option value="55574">55574</option>
                <option value="55578">55578</option>
                <option value="55579">55579</option>
                <option value="55583">55583</option>
                <option value="55585">55585</option>
                <option value="55586">55586</option>
                <option value="55587">55587</option>
                <option value="55588">55588</option>
                <option value="55589">55589</option>
              </select>
            </div>
            <div class="mb-3">
              <label for="exampleFormControlInput1" class="form-label">Name Sales</label>
              <input name="namesales" type="text" class="form-control" placeholder="Enter your Name Sales" required maxlength="100">
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.40 Code modal untuk pop up insert data di mastersales.blade

Gambar 3.40 merupakan code *modal* di *mastersales.blade* yang digunakan untuk memberikan sebuah pop up pada *insert data* dibagian *mastersales.blade*. Berfungsi untuk menampilkan sebuah form untuk mengisi form untuk tabel master sales pada admin yang dapat memperlihatkan kontak sales dan jumlah target sales. Memiliki banyak class, select dengan option untuk membuat menjadi combobox pada id sales, dan label untuk penamaan pada textbox.

Id Sales	Id User	Name Sales	Phone Number	Sales Target	Modif Date	Modif By	Action
1	55589	Luis	081297509384	30	2023-11-23 08:43:29		✎ 🗑

Showing 1 to 1 of 1 entries

Previous 1 Next

Gambar 3.41 Code tampilan tabel master sales

Gambar 3.41 merupakan code tampilan tabel master sales ini dapat menjual produksi yang ditawarkan untuk pencatatan penjualan tersebut. Pada tabel master sales berupa, id sales, id user, name sales, phone number, sales target, modif date, modif by, dan action.

14. 11 September – 16 September 2023

Pada minggu keduabelas *software developer* melakukan pengerjaan bagian master sales di bagian action edit dan delete.

```
@extends('template')
@section('content')
<div class="card-body">
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-4">
        <form action="{{ url('master/sales/edit')}}" method="post" enctype="multipart/form-data">
          @csrf
          <div class="form-group">
            <label for="">Id Sales</label>
            <input type="text" name="id" value="{{ $master->id_sales }}" disabled>
            <input type="text" name="id" value="{{ $master->id_sales }}" hidden>
            <div class="mb-3">
              <div class="input-group mb-3 mt-3">
                <span class="input-group-text w-50" id="inputGroup-sizing-default">
                  Id User</span>
                <select name="user" class="form-select" aria-label="Default select example">
                  <option value="{{ $master->id_user }}" selected>{{ $master->id_user }}</option>
                  <option value="55560">55560</option>
                  <option value="55572">55572</option>
                  <option value="55574">55574</option>
                  <option value="55578">55578</option>
                  <option value="55579">55579</option>
                  <option value="55583">55583</option>
                  <option value="55585">55585</option>
                  <option value="55586">55586</option>
                  <option value="55587">55587</option>
                  <option value="55588">55588</option>
                  <option value="55589">55589</option>
                </select>
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</section>
</div>
```

Gambar 3.42 Code mastersalesedit.blade

Gambar 3.42 merupakan code mastersalesedit.blade untuk menampilkan tulisan yang telah mengisi *insert data*. Sehingga, dapat mengubah data pada tabel ada 4 *form* yakni master sales yang berupa id user, name sales, phone number, dan sales dalam 4 *form* ini yang akan memasukkan ke dalam tabel master sales. Coding mastersalesedit.blade ada halaman berupa *template*, *form action*, dan *input*. *Template* yang dapat berfungsi untuk menampilkan halaman baru. *Form action* digunakan untuk menyimpan link pada router yang dapat membuat halaman link baru sesuai *code* yang telah dibuat. Dan *Input* yang digunakan sebagai mengatur dalam berbagai *code* dapat diatur agar data bisa menyesuaikan dengan form yang ditentukan. Memiliki banyak class, select dengan option untuk membuat menjadi combobox pada id sales, dan label untuk penamaan pada textbox.


```

<span class="input-group-text w-50" id="inputGroup-sizing-default"
  >Name Sales</span
  >
  <input
    name="namesales"
    type="text"
    class="form-control"
    value="{{ $master->vnamesales }}"
    aria-label="Sizing example input"
    aria-describedby="inputGroup-sizing-default"
    required
    maxlength="100"
  />
</div>
<div class="input-group mb-3 mt-3">
  <span class="input-group-text w-50" id="inputGroup-sizing-default"
    >Phone Number</span
    >
    <input
      name="nosales"
      type="number"
      class="form-control"
      value="{{ $master->vhp_sales }}"
      aria-label="Sizing example input"
      min="0"
      aria-describedby="inputGroup-sizing-default"
      required
      maxlength="12"
    />
  </div>

```

Gambar 3.43 Code view bagian mastersales_editblade

Gambar 3.43 merupakan code *view* bagian mastersales_editblade bagian name sales dan phone number untuk menampilkan sebuah form untuk mengedit nama sales dan nomor sales. Code mastersales_editblade ada halaman berupa memiliki banyak class, Span yang digunakan untuk memberikan pembagian pada judul form yang ditampilkan. Input yang digunakan sebagai mengatur dalam berbagai code dapat diatur agar data bisa menyesuaikan dengan form yang ditentukan, dan type untuk memilih tipe angka maupun huruf.

Id Sales		1
Id User	55589	
Name Sales		
Luis		
Phone Number	081297509384	
Sales Target	30	

Gambar 3.44 Code tampilan master sales edit

Gambar 3.44 merupakan tampilan master sales edit yang menampilkan id sales, name sales, phone number, dan sales target. Yang telah memperlihatkan kembali saat sesudah menginsertkan data yang dapat mengubah id sales, name sales, phone number, dan sales target.

```

@extends('template')
@section('content')
<div class="card-body">
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-4">
        <form action="{{ url('master/sales/delete')}}" method="post" enctype="multipart/form-data">
          @csrf
          <div class="form-group">
            <label class="me-3" for="">Are you sure delete this Sales?</label>
            <div class="form-group">
              <button type="submit" class="btn btn-warning">Delete</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
@endsection

```

Gambar 3.45 Code mastersalesdelete.blade

Gambar 3.45 merupakan code mastersalesdelete.blade untuk menampilkan yang dapat menghapus pada tabel kolom tabel master sales. Coding mastersalesdelete.blade ada halaman berupa *template*, *form action*, dan *hidden* dapat menghapus kolom data tabel master sales. *Template* pada

extends yang dapat berfungsi untuk menampilkan tampilan halaman baru. *Form action* digunakan untuk menyimpan link pada router yang dapat membuat halaman link baru sesuai *code* yang telah dibuat. Dan *hidden* digunakan saat telah menekan tombol delete maka tampilan kolom akan di hapus database akan terotomatis hapus juga.



Delete

Gambar 3.46 Code tampilan master sales delete

Gambar 3.46 merupakan tampilan master sales delete yang menampilkan sebuah pertanyaan untuk menghapus pada tabel master sales.

15. 18 September – 23 September 2023

Pada minggu ketigabelas *software developer* melakukan pengerjaan bagian validasi dari menu master sales pada insert data.

```
</div>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Name Sales</label>
  <input name="namesales" type="text" class="form-control" placeholder="Enter your Name Sales" required maxlength="100">
</div>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Phone Number</label>
  <input name="nosales" type="number" class="form-control" min="0" placeholder="Enter your Phone Number" required maxlength="12">
</div>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Sales Target</label>
  <input name="salestarget" type="number" class="form-control" placeholder="Enter Sales Target" required>
</div>
```

Gambar 3.47 Code mastersales.blade memvalidasikan bagian pop up insert data

Gambar 3.47 merupakan code mastersales.blade memvalidasikan bagian pop up insert data untuk saat melakukan pengisian form insert data jika ada tidak diisi maka ada tampilan tulisan pada setiap pengisian form yang menggunakan *required* yang berfungsi untuk membuat memastikan pengguna menginput pada insert data pada kolom yang sesuai sebelum mereka dapat mengirimkan form.

The image shows a web form titled "Insert Data Sales" with a close button (X) in the top right corner. The form contains four input fields: "Id User" (a dropdown menu with the text "Choice your Id User"), "Name Sales" (a text input field with the placeholder "Enter your Name Sales"), "Phone Number" (a text input field with the placeholder "Enter your Phone Number"), and "Sales Target" (a text input field with the placeholder "Enter Sales Target"). A red warning message box is overlaid on the "Name Sales" field, containing an exclamation mark icon and the text "Please select an item in the list." At the bottom right of the form, there are two buttons: a grey "Close" button and a blue "Save" button.

Gambar 3.48 Code tampilan pop up form insert data untuk memvalidasikan

Gambar 3.48 merupakan code tampilan pop up form insert data untuk memvalidasikan jika saat tidak melakukan pengisian form pada insert data sales maka ada tampilan tulisan yang harus mengisi.

16. 25 September – 30 September 2023

Pada minggu keempatbelas *software developer* melakukan pengerjaan bagian delete di menu master sales dan menambahkan button back pada semua menu master dibagian edit dan delete.

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

Gambar 3.49 Code button back

Gambar 3.49 merupakan code button back untuk membuat tombol ke halaman sebelumnya di karenakan pada fitur master belum ada pada button back.



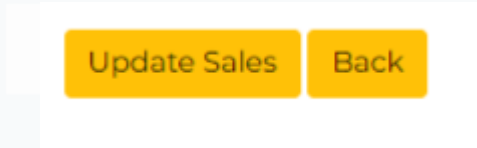
Are you sure delete this Sales?

Delete

Back

Gambar 3.50 Code tampilan tombol back pada menu master sales delete

Gambar 3.50 merupakan code tampilan tombol back bagian *delete* agar memudahkan jika ingin kembali ke tabel master sales.



Update Sales

Back

Gambar 3.51 Code tampilan tombol back pada menu master sales edit

Gambar 3.51 merupakan code tampilan tombol back bagian *edit* agar memudahkan jika ingin kembali ke tabel master sales.

17. 2 Oktober – 7 Oktober 2023

Pada minggu kelimabelas *software developer* melakukan pengerjaan bagian insert data di menu master discount untuk disabled dan *Bugs Fixing*.

```

<script>
  document.addEventListener('DOMContentLoaded', function () {
    var percentageInput = document.querySelector('[name="percentage"]');
    var valueInput = document.querySelector('[name="value"]');

    percentageInput.addEventListener('input', function () {
      var percentageValue = this.value.replace(/^[^0-9\-\]/g, '');
      if (percentageValue > 100) {
        this.value = "100";
      } else {
        this.value = percentageValue;
      }

      if (percentageValue !== "") {
        valueInput.disabled = true;
      } else {
        valueInput.disabled = false;
      }
    });

    valueInput.addEventListener('input', function () {
      var value = this.value.replace(/^[^0-9\-\]/g, '');
      this.value = value;

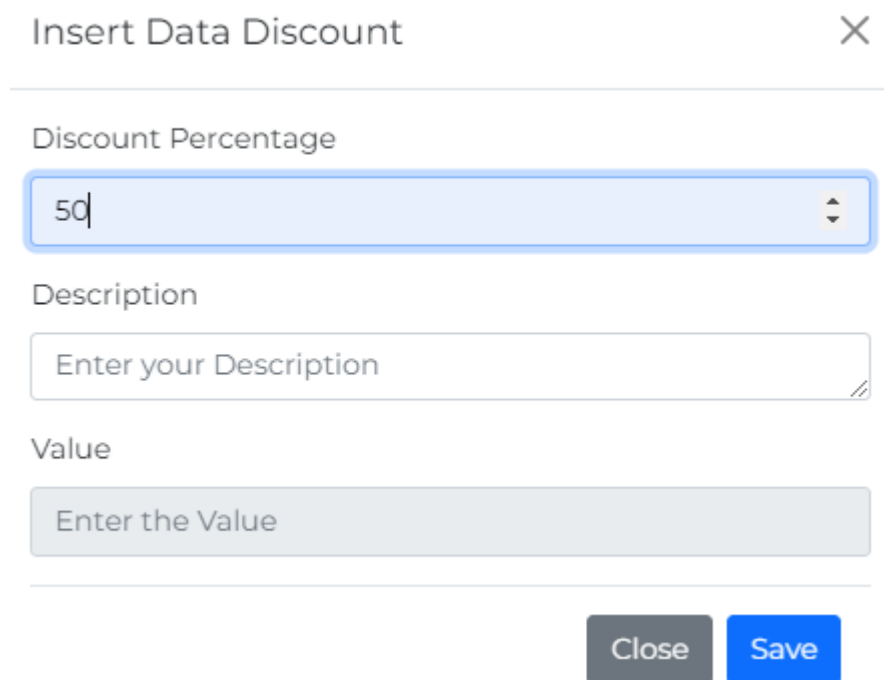
      if (value !== "") {
        percentageInput.disabled = true;
      } else {
        percentageInput.disabled = false;
      }
    });
  });
</script>

```

Gambar 3.52 Code script bagian masterdiscount.blade untuk disable percentage dan value

Gambar 3.52 merupakan code script bagian masterdiscount.blade pada *disable percentage* dan *value*. Code script ini menggunakan JQuery untuk mempermudah disable pada *percentage* dan *value*. Namun, code masterdiscount.blade ada halaman berupa *script*, *function*, dan *disable* akan mendisable agar saat menginputkan diskon bisa memilih melalui *percentage* atau *value* untuk memberikan *event* diskon tersebut. *Function* ini bagian dari program dengan nama tertentu yang digunakan untuk melakukan pekerjaan tertentu dan terpisah dari bagian program yang menggunakan fungsi tersebut. Dan *Script* yang dapat digunakan memvalidasikan data jika pada saat tidak

mengisi pada form yang ditampilkan maka ada sebuah munculnya tulisan yang wajib diisi oleh pengguna tersebut.



Insert Data Discount

Discount Percentage

50

Description

Enter your Description

Value

Enter the Value

Close Save

Gambar 3.53 Code tampilan disable pada percentage dan value

Gambar 3.53 merupakan code tampilan yang akan mendisable agar saat menginputkan diskon bisa memilih melalui *percentage* atau *value* untuk memberikan *event* diskon tersebut. Dalam *bugs fixing* ini *Quality assurance* dapat mengecek apakah ada kendala *error*, kendala salah penempatan, maupun warna agar kendala yang sudah ditemukan akan di perbaiki lagi oleh *developer*.

18. 9 Oktober – 14 Oktober 2023

Pada minggu keenambelas *software developer* melakukan pengerjaan bagian menggantikan warna button back orange menjadi warna biru dan *Bugs Fixing*.

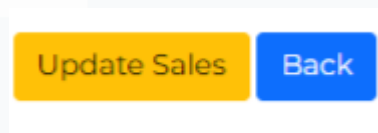
```

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

```

Gambar 3.54 Code button back menjadi warna biru

Gambar 3.54 merupakan code button back menjadi warna biru yang menggantikan class btn-primary agar button back berubah menjadi warna biru.



Gambar 3.55 Code tampilan button back sudah menjadi warna biru

Gambar 3.55 merupakan code tampilan button back yang sudah merubah warna menjadi warna biru dikarenakan PMO PT Ritzproject ingin menggantikan warna biru agar bisa membedakan warna pada button tersebut.

19. 16 Oktober – 27 Oktober 2023

Pada minggu ketujuhbelas dan kedelapanbelas *software developer* melakukan pengerjaan merevisikan menu master bagian admin di delete, menambahkan button back yang belum ada, dan *bugs fixing*.

```

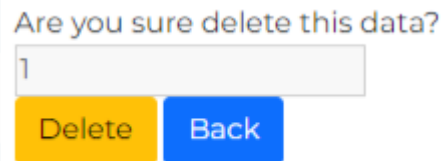
<!-- Menghubungkan dengan view template master -->
@extends('template')
@section('content')
<div class="card-body">
<div class="container-fluid">
<div class="row">
<div class="col-md-4">
<form action="{{ url('master/discount/delete')}}" method="post" enctype="multipart/form-data">
@csrf
<div class="form-group">
<label class="me-3" for="">Are you sure delete this data?</label>

<input type="text" name="id" value="{{ $id }}" disabled>
<input type="text" name="id" value="{{ $id }}" hidden>
</div>
<div class="form-group">
<button type="submit" class="btn btn-warning">Delete</button>
<button type="button" onclick="history.back();" class="btn btn-primary">Back</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Gambar 3.56 Code masterdiscountdelete.blade untuk merevisikan pada delete

Gambar 3.56 merupakan code `masterdiscountdelete.blade` untuk merevisikan pada `delete` agar menampilkan id diskon. Code `masterdiscountdelete.blade` ada halaman bagian merevisikan yakni *value* untuk menampilkan id, *disable*, dan *hidden* untuk menghapusnya. *Disable* ini berfungsi untuk menonaktifkan sebuah form yang tidak dapat diisi atau mengubah Kembali. Dalam *bugs fixing* ini *Quality assurance* dapat mengecek apakah ada kendala *error*, kendala salah penempatan, maupun warna agar kendala yang sudah ditemukan akan di perbaiki lagi oleh *developer*.



Gambar 3.57 Code tampilan delete sudah direvisikan yang menampilkan id discount

Gambar 3.57 merupakan code tampilan delete sudah direvisikan yang menampilkan id discount dan semua menu masternya juga merevisikan seperti master diskon delete.



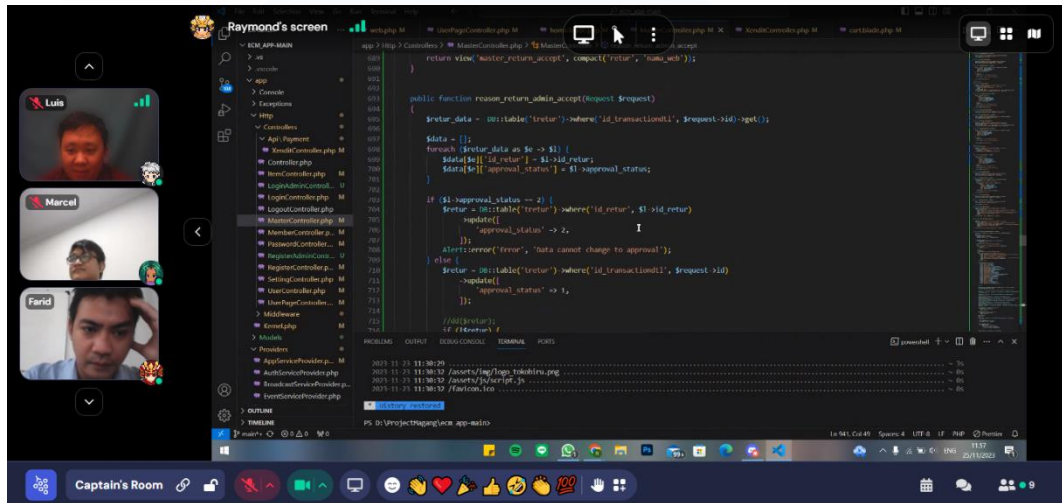
Gambar 3.58 Code tampilan button back untuk user bagian checkout barang

Gambar 3.58 ini *software developer* menambahkan *button back* yang belum ada untuk *user* bagian *checkout* jika pengguna ingin kembali.

20. 31 Oktober – 24 November 2023

Pada minggu kesembilanbelas sampai keduapuluhdua *software developer* sudah menyelesaikan *task-task* yang diberikan oleh PMO PT Ritzproject Sinergi Visitama. Dalam fase *Phase Feedback UAT Result* ini *software developer* menunggu hasil *feedback* oleh PMO PT Ritzproject

Sinergi Visitama untuk melakukan penyelesaian aplikasi berbasis website Ritzapps apakah bisa *go live* apa masih belum.



Gambar 3.59 Meeting terakhir untuk foto dengan tim dan PMO PT Ritzproject Sinergi Visitama

Gambar 3.59 ini merupakan meeting terakhir untuk berkerja magang di PT Ritzproject Sinergi Visitama dengan tim dan PMO PT Ritzproject. Yang menjadi sebuah moment terakhir magang Ritzproject di batch kedua ini bisa menjadikan pengalaman yang baru saat memulai periode magang sampai mengakhiri periode magang ini. Dan penulis ingin menyampaikan terima kasih kepada Bapak Farid Ananda yang selaku pembimbing lapangan dan PMO PT Ritzproject Sinergi Visitama.yang selalu mendampingi saat penulis magang, memberikan ilmu untuk pembekalan saat di dunia kerja setelah lulus kuliah, dan memiliki pengalaman yang berharga dengan waktu berkerja magang selama periode magang.

3.3 Kendala yang Ditemukan

Ketika mahasiswa sedang menjalani masa magang, seringkali mereka menghadapi tantangan atau hambatan saat melaksanakan tugas yang diberikan oleh supervisor. Berikut adalah beberapa kesulitan yang umumnya dihadapi oleh mahasiswa selama kegiatan magang:

1. Salah satu indikasi konkret dari kesulitan tersebut adalah ketika mahasiswa dihadapkan pada penggunaan alat pengelola basis data yang baru bagi

mereka, seperti DBeaver dan Laravel. Kendala ini dapat menyebabkan keterlambatan dalam pekerjaan karena mahasiswa perlu beradaptasi dengan alat tersebut sebelum dapat efektif bekerja.

2. Kendala kedua melibatkan kesulitan yang dihadapi oleh mahasiswa selama menjalani magang secara *work from home*. Kesulitan ini dapat menghambat komunikasi, terutama ketika mahasiswa berpartisipasi dalam pertemuan virtual. Seringkali, terdengar suara-suara gangguan yang dapat mengganggu kualitas suara pembicara. Kendala ini juga berpotensi memperlambat kemajuan kerja, karena gangguan tersebut bisa menghambat aliran informasi terkait tugas yang sedang dijalankan.

3.4 Solusi atas Kendala yang Ditemukan

Setiap kali mahasiswa menghadapi masalah atau kendala selama pelaksanaan kegiatan magang, mereka berupaya untuk menemukan solusi yang sesuai. Berikut adalah beberapa solusi yang dapat diterapkan untuk mengatasi setiap kendala tersebut yakni:

1. Untuk mengatasi kendala pertama, mahasiswa berusaha mencari tutorial melalui platform seperti YouTube dan Google. Tujuannya adalah untuk memperoleh informasi tambahan yang dapat meningkatkan pemahaman terkait penggunaan alat, khususnya aplikasi DBeaver. Dengan cara ini, diharapkan kendala dalam menggunakan DBeaver dan Laravel dapat diminimalkan.
2. Untuk mengatasi kendala kedua, yaitu ketidaknyamanan dalam pertemuan online selama kegiatan magang, mahasiswa mengambil langkah dengan memberi tahu supervisor bahwa terdapat gangguan suara, seperti kebisingan yang mengganggu, selama pertemuan online. Setelah mengetahui situasi tersebut, pembicara dapat memilih untuk mengubah lokasi atau

menggunakan perangkat pendengar, seperti headphone atau headset, yang dapat mengurangi suara gangguan.

