

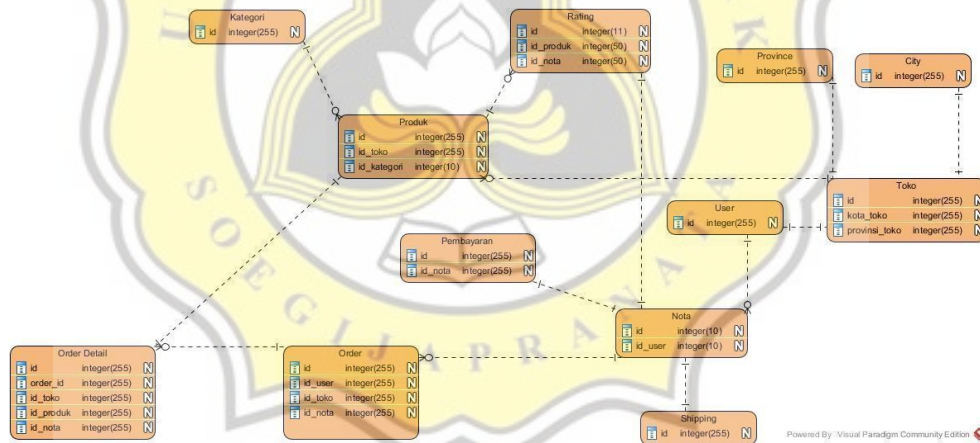
## BAB IV

### PEMBAHASAN

#### 4.1. Perancangan Aplikasi Website Dcreatif Unika

##### 4.1.1. Rancangan Database (Entity Relationship Diagram)

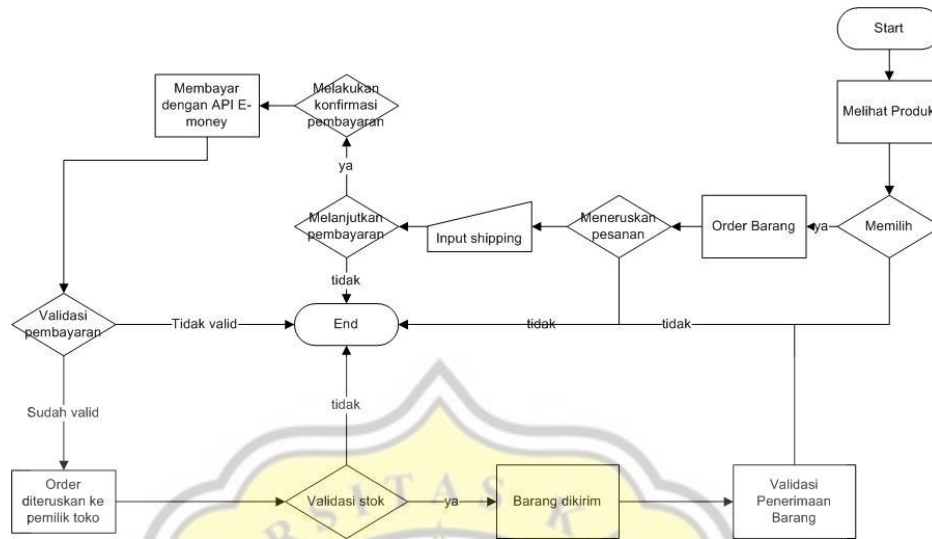
Sebelum membuat website Dcreatif Unika, perlu melakukan perancangan database. Setiap tabel pada database perlu memiliki *Primary Key* maupun *Foreign Key* yang nantinya akan digunakan untuk merelasikan antar tabel. Fungsi dari database ini untuk merekam semua transaksi yang ada pada website Dcreatif Unika. Pada rancangan database yang digunakan di Dcreatif Unika terdapat pada gambar 4.1.



Gambar 4.1 Entity Relationship Database

##### 4.1.2. Flowchart

Flowchart ini merupakan desain untuk mengetahui alur sistem website yang ada di Dcreatif Unika Flowchart ini dapat dilihat pada gambar 4.2.

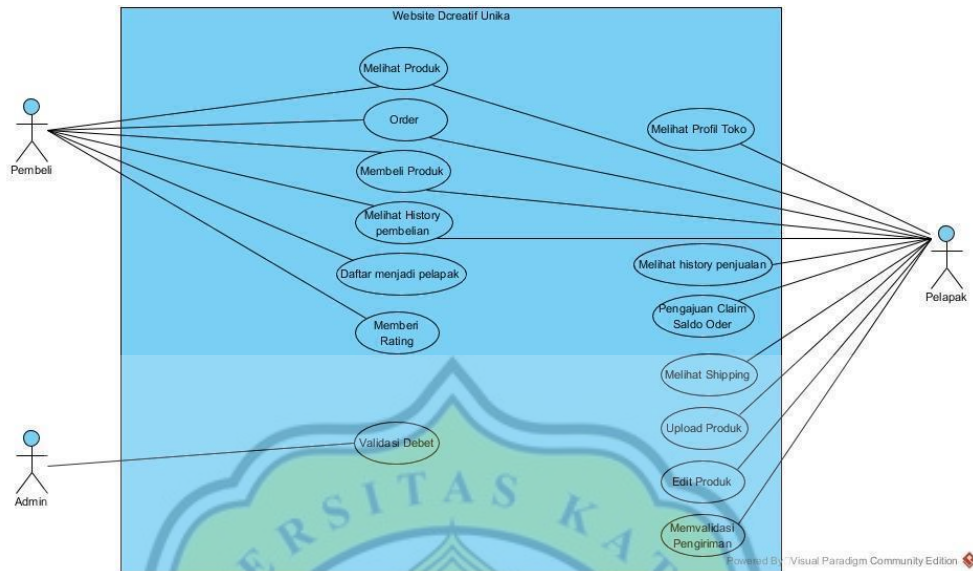


Gambar 4.2 Flowchart Diagram

Alur proses pada Dcreatif Unika adalah pertama konsumen melihat produk, kemudian masukkan produk ke dalam keranjang, kemudian memasukan detail pengiriman, baru melakukan checkout. Setelah melakukan checkout sistem akan mencetak nota dan konsumen melakukan pembayaran menggunakan pembayaran yang telah disediakan, setelah membayar sistem akan otomatis memperbaharui data di database, kemudian pemilik toko akan mendapatkan notifikasi, kemudian pemilik toko akan menerima orderan, kemudian memasukan resi pengiriman, setelah barang sampai di rumah konsumen, konsumen dapat memberikan rating produk pada toko tersebut.

#### 4.1.3. Use Case Diagram

Use Case Diagram ini untuk mengakses fitur sesuai dengan tiap role di website. Use Case Diagram Dcreatif Unika ini dapat dilihat pada gambar 4.3.



Gambar 4.3 Use Case Diagram

Dalam rancangan ini terdapat tiga aktor, yang melakukan interaksi di dalam sistem, antara lain: pembeli, pelapak, dan admin. Masing-masing aktor dapat mengakses fitur sesuai dengan role masing-masing. Pembagian role ini nanti akan diatur sesuai email yang didaftarkan setiap user. Pelapak ini bisa melihat produk, melihat profil toko, melihat history penjualan, mengajukan *claim* saldo, melihat shipping, mengupload produk, mengedit produk, memvalidasi pengiriman. Kalau pembeli bisa melihat produk, melakukan order, membeli produk, melihat history pembelian, mendaftar menjadi pelapak, dan memberikan rating. Sedangkan admin ini adalah pengelola website Dcreatif Unika yang berfungsi untuk memvalidasi debet dari pemilik toko.

## 4.2. Pembuatan Website

### 4.2.1. Pembuatan Database MySql

Untuk membuat database Dcreative Unika ini, menggunakan DBMS (Database Management System) MySql. Database ini merupakan software gratis yang bisa digunakan oleh siapa saja. Menggunakan database ini karena MySql terintegrasi dengan bahasa pemrograman PHP. Desain database Dcreatif Unika dapat kita lihat pada gambar 4.4.



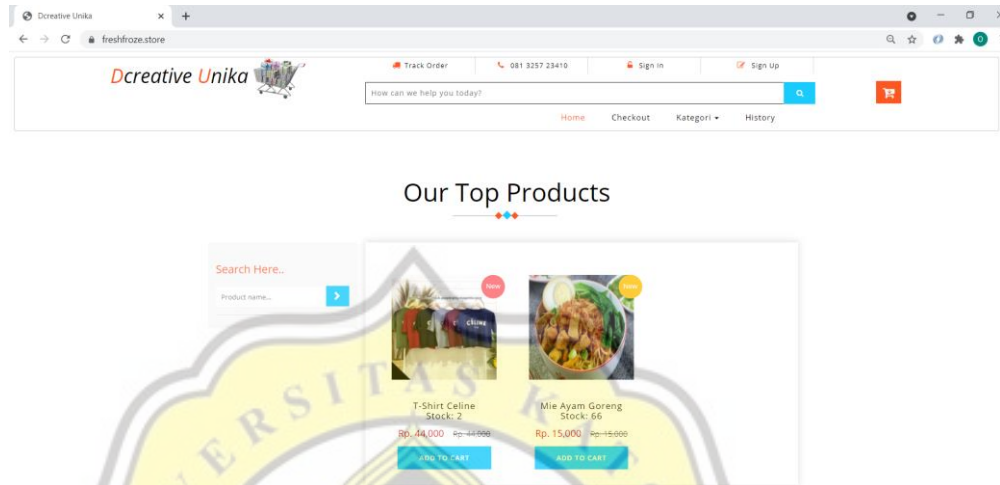
Gambar 4.4 Tampilan design database MySql

### 4.2.2. Tampilan Website

#### 4.2.2.1. Halaman Konsumen

Halaman awal ini bisa diakses tanpa harus login, agar konsumen bisa melihat-lihat barang terlebih dahulu. Halaman awal ini berisi produk-produk yang dijual oleh pemilik toko. Setiap toko memiliki 1 produk yang akan dijadikan display di halaman awal secara random. Tampilan awal Website

Dcreatif Unika sebelum login dapat kita lihat pada gambar 4.5.



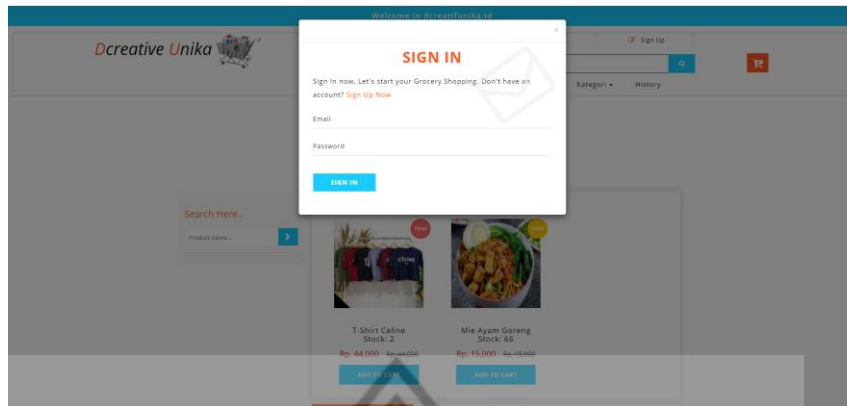
Gambar 4.5 Tampilan awal Website Dcreatif Unika sebelum login

Untuk halaman awal konsumen, semua data produk akan ditampilkan dalam variabel \$produk kemudian akan dilakukan perulangan hingga semua produk tampil sesuai dengan data yang ada di database. *Source Code* dapat dilihat pada gambar 4.6.

```
@foreach($kategoris as $kategori)
<?php
$produk =\App\Models\Produsen::where('kategori',$kategori->nama_kategori)->where('tampil',0)->orderBy('created_at','desc')
?>
@foreach($produk as $produk)
<div class="col-md-4 product-men">
  <div class="men-pro-item simpleCart_shelfItem">
    <div class="men-thumb-item">
      
      <div class="men-cart-pro">
        <div class="inner-men-cart-pro">
          <a href="{{url('pesan')}}{{$produk->id}}" class="link-product-add-cart">Quick View</a>
        </div>
      </div>
      <span class="product-new-top">New</span>
    </div>
    <div class="item-info-product ">
      <h4>
        <a href="single.html">{{$produk->nama_produk}}</a><br/>
        <a href="single.html">Stock: {{$produk->stock}}</a>
      </h4>
      <div class="info-product-price">
        <span class="item_price">Rp. {{number_format($produk->harga)}}</span>
        <del>Rp. {{number_format($produk->harga)}}</del>
      </div>
      <div class="snipcart-details top_brand_home_details item_add single-item hvr-outline-out">
        <form action="{{url('pesan')}}{{$produk->id}}" method="get">
          <fieldset>
            <input type="submit" name="submit" value="Add to cart" class="button" />
          </fieldset>
        </form>
      </div>
    </div>
  </div>
</div>
```

Gambar 4.6 Source Code tampilan awal Dcreatif Unika

Formulir login ini berupa modal show (modal show adalah jendela kecil yang diundang oleh javascript diatas halaman utama sehingga tidak perlu memindah halaman), setelah memasukan email dan password secara benar modal ini akan otomatis menutup dan diarahkan ke halaman utama (Sistem akan memvalidasi email dan password berdasarkan data yang direkam di *database*. Namun, ketika email dan password yang dimasukan salah maka terdapat pemberitahuan email atau password salah. Formulir login ini dapat kita lihat pada gambar 4.6.



Gambar 4.6 Formulir login Dcreatif Unika

Formulir login Dcreatif Unika ini menggunakan email dan password yang telah didaftarkan di Dcreatif Unika, kemudian sistem akan memvalidasi email dan password dengan data di database. *Source Code* formulir login Dcreatif Unika dapat dilihat pada gambar 4.7.

```

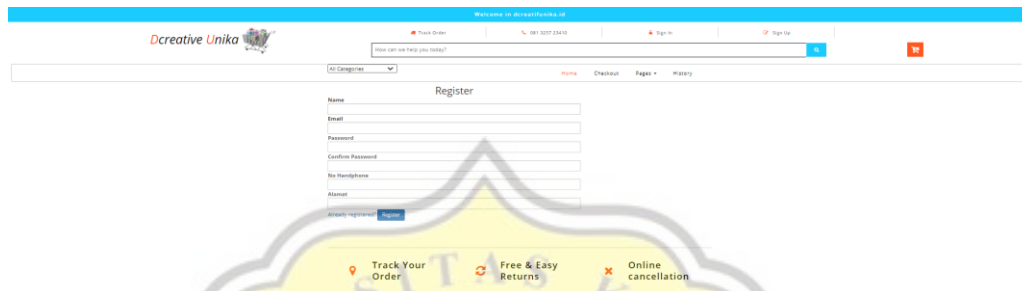
<div class="modal-content">
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal">&times;</button>
  </div>
  <div class="modal-body modal-body-sub_agile">
    <div class="main-mailposi">
      <span class="fa fa-envelope-o" aria-hidden="true"></span>
    </div>
    <div class="modal-body-left modal-body-left1">
      <h3 class="agileinfo_sign">Sign In </h3>
      <p>
        Sign In now, Let's start your Grocery Shopping. Don't have an account?
        <a href="{{url('register')}}" >
          Sign Up Now</a>
      </p>
      <form action="{{ route('login') }}" method="post">
        {{csrf_field()}}
        <div class="styled-input agile-styled-input-top">
          <input type="email" placeholder="Email" name="email" required="">
        </div>
        <div class="styled-input">
          <input type="password" placeholder="Password" name="password" required="">
        </div>
        <input type="submit" value="Sign In">
      </form>
      <div class="clearfix"></div>
    </div>
    <div class="clearfix"></div>
  </div>
</div>

```

Gambar 4.7 *Source Code* formulir login Dcreatif Unika

Formulir register ini berfungsi untuk mendaftarkan akun ke Dcreatif Unika untuk pertama kali. Data yang disimpan untuk pendaftaran akun ini adalah nama pengguna, email, password dan password confirmation, nomor handphone, dan alamat. Setelah berhasil mendaftarkan akun, setiap pengguna wajib melakukan validasi email dengan

membuka email. Setelah melakukan validasi baru akun akan aktif. Formulir register akun Dcreatif Unika dapat kita lihat pada gambar 4.8.



Gambar 4.8 Formulir registrasi akun Dcreatif Unika

Formulir register ini terdapat beberapa validasi tipe data dan jumlah karakter. Kemudian untuk password akan diubah menjadi kode sebelum didaftarkan di database. *Source Code* validasi registrasi akun Dcreatif Unika dapat dilihat pada gambar 4.9

```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'no_hp' => ['required', 'string', 'max:15'],
        'alamat' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:50', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'no_hp' => $data['no_hp'],
        'email' => $data['email'],
        'role' => "konsumen",
        'alamat' => $data['alamat'],
        'password' => Hash::make($data['password']),
    ]);
}
```

Gambar 4.9 *Source Code* validasi registrasi akun Dcreatif Unika



Setelah login, menu sign in dan sign out akan hilang dan digantikan oleh logout. Kemudian jika akun tersebut memiliki pesanan di dalam keranjang maka akan muncul angka di dekat notifikasi keranjang dengan angka sesuai dengan jumlah produk yang dipesan. Jika akun tersebut belum menjadi pemilik toko maka notifikasi untuk daftar pesanan toko akan hilang, namun jika sudah menjadi pemilik toko notifikasi daftar pesanan toko akan muncul beserta jumlah daftar pesanan di sebelah kanannya. Tampilan awal Website Dcreatif Unika setelah login dapat kita lihat pada gambar 4.10.



Gambar 4.10 Tampilan awal Website Dcreatif Unika setelah login

Halaman detail produk ini berfungsi untuk melihat detail barang yang akan dibeli oleh konsumen. Konsumen juga dapat melihat review dan rating dari produk yang akan dibeli. Selain itu, konsumen juga dapat melihat jumlah stok yang tersisa dari produk yang dibeli tersebut. Pelapak dapat mengupload detail produk pada halaman informasi toko. Halaman detail produk dapat kita lihat pada gambar 4.11.



Gambar 4.11 Halaman detail produk

Halaman detail ini mengambil data untuk keterangan/detail produk, rating produk beserta komentar pembeli. *Source Code* halaman detail produk dapat kita lihat pada gambar 4.12

```

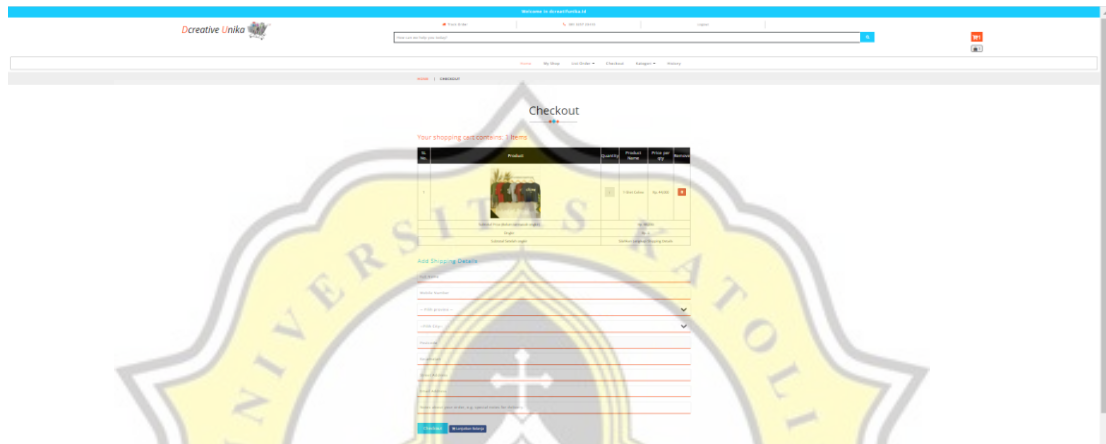
<div class="col-md-7 single-right-left simpleCart_shelfItem">
  <h3>{{ $produk->nama_produk }}</h3>
  <div class="rating1">
    <span class="starRating">
      <?php
        $ratingget = \App\Models\rating::where('id_produk',$produk->id)->get();
        $jumlah = \App\Models\rating::where('id_produk',$produk->id)->count();
        $rating = \App\Models\rating::where('id_produk',$produk->id)->avg('nilai');
        $parse = floor($rating);
        $int = (int)$parse;
        $img="";
        $i=1;
        while($i<=$int){
          $img = $img."<i class='fa fa-star' style='color:yellow;'></i>";
          $i=$i+1;
        }
        for($i=$int+1; $i<=5; $i++){
          $img = $img."<i class='fa fa-star'></i>";
        }
        echo $img;
      ?>
    </span>
    {{ $jumlah." (Reviews) }}
  </div>

```

Gambar 4.12 *Source Code* Halaman Detail Produk

Halaman checkout ini berisi daftar barang yang akan dibeli dan alamat tujuan pengiriman. Jadi sebelum konsumen

melakukan checkout, konsumen harus mengisi alamat pengiriman dengan lengkap. Pertama kali konsumen harus menginputkan data ke dalam database. Alamat tersebut dapat digunakan lagi setelah konsumen melakukan order baru. Halaman checkout dapat kita lihat pada gambar 4.13.



Gambar 4.13 Halaman *checkout*

Pada halaman *checkout* ini untuk data provinsi, kota, dan kode pos diambil dari database Rajaongkir yang transfer datanya menggunakan javascript dan JSON. *Source Code* halaman *checkout* dapat kita lihat pada gambar 4.14

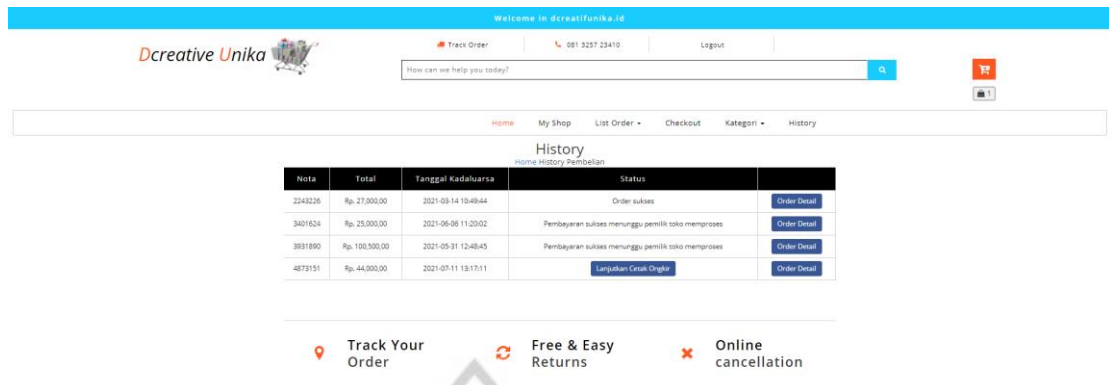
```

<form action="{{url('konfirmasicheckout')}}" method="post" class="credittly-card-form agileinfo_form">
  {{csrf_field()}}
  <div class="credittly-wrapper wthree, w3_agileits_wrapper">
    <div class="information-wrapper">
      <div class="first-row">
        <div class="controls">
          <input class="billing-address-name" type="text" name="namapenerima" placeholder="Full Name" required="">
        </div>
        <div class="w3_agileits_card_number_grids">
          <div class="w3_agileits_card_number_grid_left">
            <div class="controls">
              <input type="text" placeholder="Mobile Number" name="nohp" required="">
            </div>
          </div>
          <div class="clear"> </div>
        </div>
        <input class="billing-address-name" type="hidden" name="provinceinput" placeholder="Full Name" required="">
        <input class="billing-address-name" type="hidden" name="cityidinput" placeholder="Full Name" required="">
        <input class="billing-address-name" type="hidden" name="citynamainput" placeholder="Full Name" required="">
        <input class="billing-address-name" type="hidden" name="tipeinput" placeholder="Full Name" required="">
        <div class="controls">
          <select class="option-w3ls" name="destinationprovinsi" required="">
            </select>
        </div>
        <div class="controls">
          <select class="option-w3ls" name="kota_tujuan" required="">
            <option value=""> --Pilih City-- </option>
          </select>
        </div>
        <div class="w3_agileits_card_number_grid_right">
          <div class="controls">
            <input type="text" placeholder="Postcode" name="kodepos" required="" disabled="">
            <input type="hidden" placeholder="Postcode" name="kode_pos" required="" >
          </div>

```

Gambar 4.14 *Source Code* halaman *checkout*

Halaman ini berfungsi untuk melihat pesanan yang sedang berjalan maupun yang sedang diproses untuk konsumen. Selain itu juga memberikan informasi tanggal pemesanan produk, kode nota, dan status dari transaksi tersebut. Halaman riwayat pembelian dapat kita lihat pada gambar 4.15.



Gambar 4.15 Halaman riwayat pembelian

Riwayat pembelian konsumen ini berupa tabel yang barisnya *dilooping* hingga semua data tampil. *Source code* halaman riwayat pembelian dapat kita lihat pada gambar 4.16

```

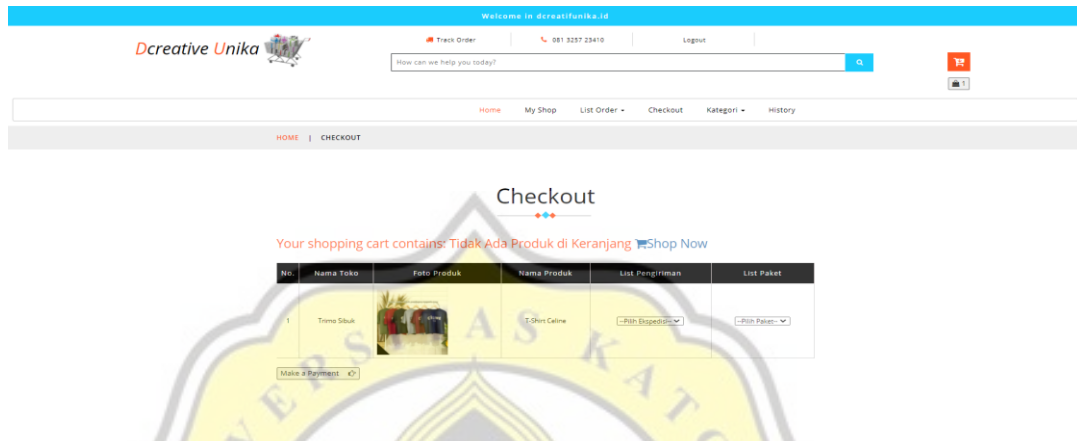
<table class="timetable_sub">
  <thead>
    <tr>
      <th>Nota</th>
      <th>Total</th>
      <th>Tanggal Kadaluarsa</th>
      <th>Status</th>
    </tr>
  </thead>
  <tbody>
    @if(!empty($nota))
    @foreach($nota as $nota)
      <tr>
        <td>
          {{ $nota->id }}
        </td>
        <td>
          Rp. {{ number_format($nota->total_pembayaran+$nota->ongkir) }},00
        </td>
        <td>
          {{ $nota->payment_due }}
        </td>
        <td>
          @if($nota->proses==0)
            Silahkan menyelesaikan order
          @elseif($nota->proses==1)
            <a href="{{url('/pilihkekspedisi')}}{{ $nota->id }}" class="btn btn-primary">Lanjutkan Cetak Ongkir</a>
          @elseif($nota->proses==2)
            <a href="{{ $nota->payment_url }}" class="btn btn-primary">Lanjutkan Pembayaran</a>
          @elseif($nota->proses==3)
            @if($nota-> kirim==0)
              Pembayaran sukses menunggu pemilik toko memproses
            </if>
          </td>
      </tr>
    @endforeach
  </tbody>
</table>

```

Gambar 4.16 *Source Code* halaman riwayat pembelian

Setelah melakukan checkout maka konsumen harus memilih ekspedisi yang dipakai untuk mengirim barang, beserta paket yang tersedia pada ekspedisi tersebut untuk mencetak ongkos kirim. Ongkos kirim akan diambil melalui

API Rajaongkir dengan mengirimkan kota asal, kota tujuan, dan paket yang dipilih. Halaman memilih ekspedisi dan mencetak ongkir dapat kita lihat pada gambar 4.17.



Gambar 4.17 Halaman memilih ekspedisi dan mencetak ongkir

Pemilihan ekspedisi ini menggunakan tag *select* seperti biasa sesuai akun gratis yang disediakan oleh Rajaongkir yaitu Pos Indonesia, Tiki, dan JNE. Sedangkan untuk paket ini mengirimkan data kota asal (pemilik toko), kota tujuan (pembeli), berat dari produk, dan ekspedisi yang digunakan. Setelah data semua dikirimkan dengan JSON kemudian akan ada balikan dari server Rajaongkir dan diubah dalam bentuk list *dropdown*. *Source code* halaman pemilihan paket dapat kita lihat pada gambar 4.18.

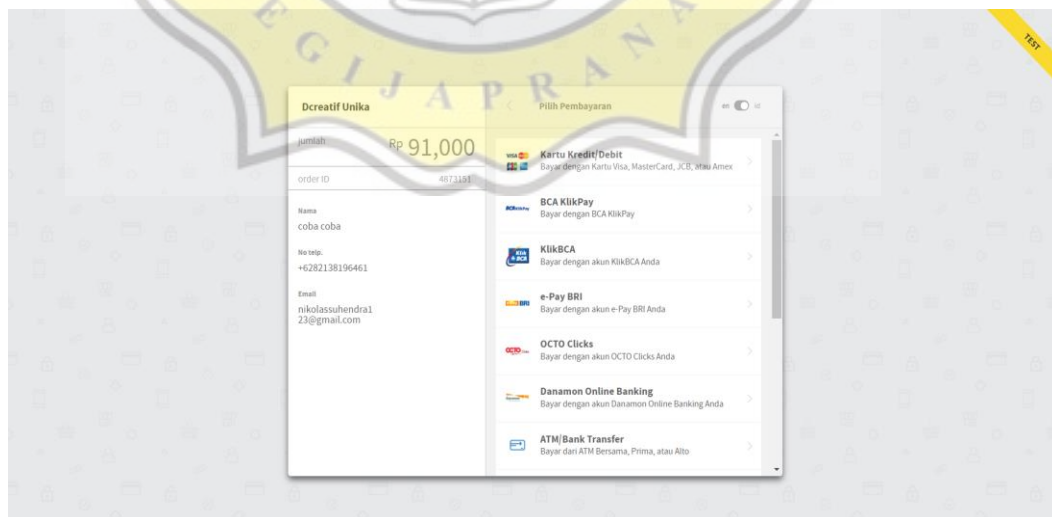
```

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    $array_response = json_decode($response, TRUE);
    $paket = $array_response["rajaongkir"]["results"]["0"]["costs"];
    echo "<option value='>-- Pilih paket --</option>";
    foreach($paket as $key => $tiappaket){
        echo "<option
        paket='".$tiappaket["service"]."'
        ongkir='".$tiappaket["cost"]["0"]["value"]."'
        etd='".$tiappaket["cost"]["0"]["etd"]."'
        >";
        echo $tiappaket["service"]." ";
        echo "Rp. ".number_format($tiappaket["cost"]["0"]["value"])." ";
        echo $tiappaket["cost"]["0"]["etd"]." HARI";
        echo "</option>";
    }
}

```

Gambar 4.18 *Source code* halaman pemilihan paket

Setelah ongkir diketahui, nota otomatis akan dibuat oleh sistem dan selanjutnya konsumen harus melakukan pembayaran melalui API Midtrans yang sudah tersedia. Midtrans menyediakan banyak metode pembayaran, dari bank transfer, e-money, pembayaran melalui indomaret dan alfamart, kartu kredit, dan masih banyak lagi. Halaman pembayaran Midtrans dapat kita lihat pada gambar 4.19.



Gambar 4.19 Halaman pembayaran Midtrans

Karena pembayaran menggunakan Midtrans, kita perlu mengirimkan data konsumen kita kepada Midtrans untuk membuat transaksi penjualan yang nantinya bisa diproses metode pembayarannya menggunakan apa sesuai yang sudah disediakan Midtrans. *Source code generate* pembayaran Midtrans dapat kita lihat pada gambar 4.20.

```
private function _generatePaymentToken($nota)
{
    $user= User::where('id',$nota->id_user)->first();
    $this->initPaymentGateway();

    $customerDetails = [
        'first_name' => $user->name,
        'last_name' => $user->name,
        'email' => $user->email,
        'phone' => $user->no_hp,
    ];

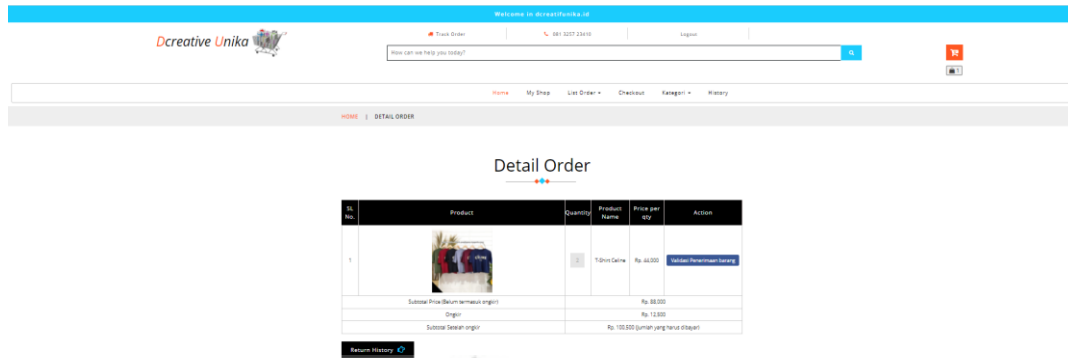
    $params = [
        'enable_payments' => \App\Models\Payment::PAYMENT_CHANNELS,
        'transaction_details' => [
            'order_id' => $nota->id,
            'gross_amount' => $nota->total_pembayaran+$nota->ongkir,
        ],
        'customer_details' => $customerDetails,
        'expiry' => [
            'start_time' => date('Y-m-d H:i:s T'),
            'unit' => \App\Models\Payment::EXPIRY_UNIT,
            'duration' => \App\Models\Payment::EXPIRY_DURATION,
        ],
    ];

    $snap = \Midtrans\Snap::createTransaction($params);
}
```

Gambar 4.20 *Source code generate* pembayaran midtrans

Setelah barang sampai di tempat tujuan, konsumen harus melakukan validasi barang sudah sampai dengan melihat di halaman history konsumen. Setelah barang diterima oleh konsumen maka uang yang ditransfer oleh konsumen ke Dcreatif Unika akan diteruskan kepada pemilik toko. Validasi penerimaan barang dapat kita lihat pada gambar 4.21.





Gambar 4.21 Validasi penerimaan barang

Tombol validasi penerimaan barang ini menggunakan halaman detail order, namun diberi syarat sesuai dengan proses yang berjalan. Jika konsumen pada tahap, pemilik mengirim barang maka tombol yang muncul adalah validasi penerimaan barang. Jika konsumen pada tahap sudah menerima barang, maka tombol yang muncul adalah pemberian rating. *Source code* validasi penerimaan barang dan pemberian rating dapat dilihat pada gambar 4.22.

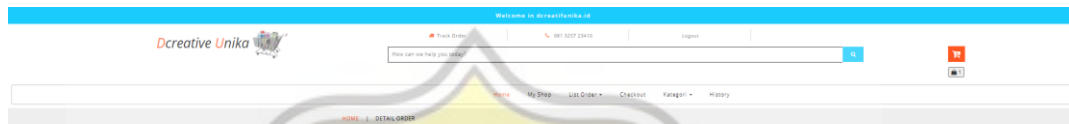
```

<?php
$cekorder= \App\Models\Order::where('id',$pesanan_detail->order_id)->first();
//dd($cekorder);
?>
<td class="invert">
    @if($cekorder->kirim==0)
        @if($nota_order->payment_status=="unpaid")
            silahkan selesaikan order
        @else
            tunggu pemilik toko memvalidasi pesanan anda
        @endif
    @else
        @if($pesanan_detail->terima==0)
            <a href="{{url('validasiterima')}}/{{$pesanan_detail->id}}" class="btn btn-primary">Validasi Penerimaan barang</a>
        @elseif($pesanan_detail->terima==1)
            @if($pesanan_detail->selesai==0)
                <a href="#" onclick="rating({{{$pesanan_detail->id_produk}},{{$pesanan_detail->id}});" class="btn btn-info mb-2"><i class="fa fa-star
            @else
                Done
            @endif
        @else
            Done
        @endif
    @endif
</td>
</tr>

```

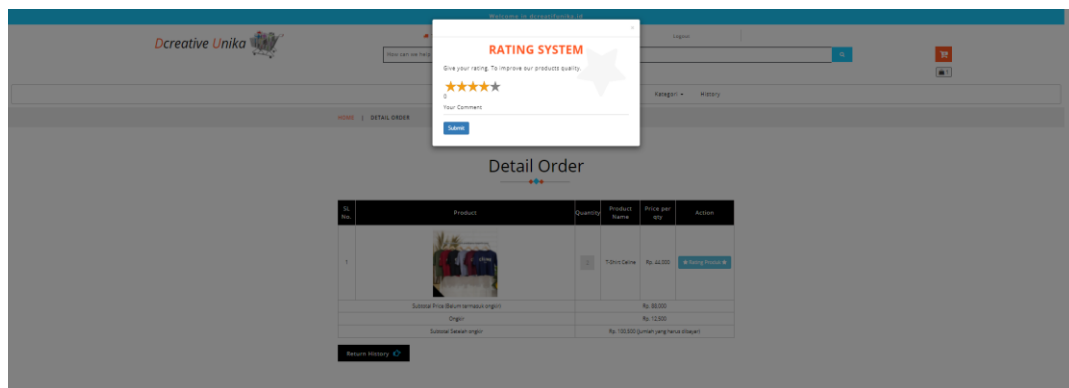
Gambar 4.22 *Source Code* validasi penerimaan barang dan pemberian rating

Setelah konsumen melakukan validasi penerimaan barang, maka konsumen wajib memberikan penilaian produk yang dibeli dan memberikan komentar terhadap produk yang dibeli. Penilaian ini nantinya akan menjadi review di halaman detail produk. Halaman pemberian penilaian produk dapat kita lihat pada gambar 4.23




Gambar 4.23 Halaman pemberian penilaian produk


Penilaian produk ini dengan melakukan klik pada bintang yang tersedia, dan memasukan komentar di kolom komentar. Setelah konsumen menekan tombol submit maka transaksi dianggap sudah selesai dan status akan berubah menjadi pesanan berhasil. Modal Show penilaian produk dapat kita lihat pada gambar 4.24



Gambar 4.24 Modal Show penilaian produk

Sebelum mencetak nota konsumen harus mengisi data ekspedisi yang akan digunakan dan paket yang dipilih. Hal ini bertujuan untuk menentukan jumlah ongkos kirim yang nantinya akan dicetak pada nota. Ongkos kirim ini sesuai dengan data yang ada di ekspedisi yang dipilih tersebut karena dengan API (*Application Programming Interface*) ini langsung mengecek ke database pihak ekspedisi dan bisa berubah-ubah secara *realtime*. Tampilan data yang diambil dari API Raja Ongkir dapat kita lihat pada gambar 4.25.

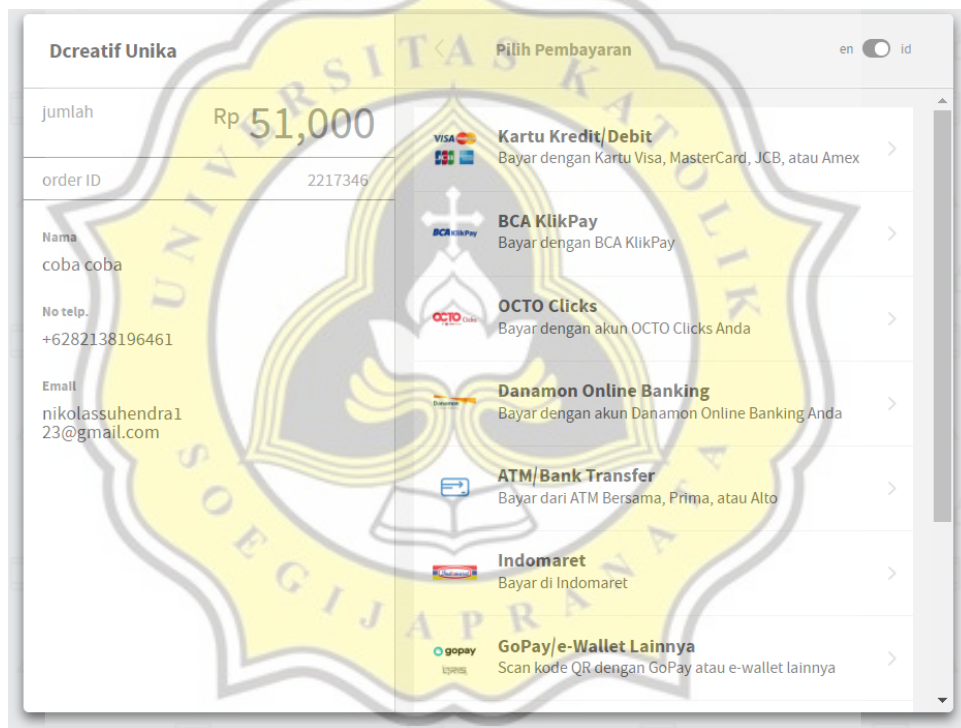
No.	Nama Toko	Foto Produk	Nama Produk	List Pengiriman	List Paket
1	Trimo Sibuk		T-Shirt Celine	--Pilih Ekspedisi--	--Pilih Paket--

Make a Payment 

Gambar 4.25 Data dari API Raja Ongkir

Setelah semua data pengiriman lengkap maka selanjutnya meneruskan mencetak nota dan mengirimkan data konsumen ke Midtrans agar dibuatkan transaksi menggunakan API. Setelah konsumen membayar maka otomatis Midtrans akan mengirimkan notifikasi ke Dcreatif Unika untuk memvalidasi pembayaran pada nota yang sudah dicetak tadi secara otomatis. Sehingga transaksi ini pembayarannya akan divalidasi secara *realtime* jika konsumen sudah benar membayar dan terverifikasi oleh sistem.

Karena menggunakan Midtrans sebagai pihak ke-3 maka terdapat biaya untuk 1 kali transaksi. Untuk bank transfer dalam 1 transaksi dikenakan biaya Rp. 4.000,00. Biaya ini untuk orang umum yang menggunakan. Ketika bekerja sama dengan pihak Midtrans maka transaksi ini bisa berkurang dan bisa dibicarakan dengan pihak sales Midtrans. Contoh tampilan transaksi menggunakan API Midtrans dapat kita lihat pada gambar 4.26.



Gambar 4.26 Data dari API Midtrans

Untuk melakukan konfigurasi ke rajaongkir, menggunakan CURL untuk melakukan inisialisasi mendapatkan halaman website yang dituju. Sesuai dengan alamat URL <https://api.rajaongkir.com/starter/province>. Kemudian kita juga harus mengirimkan key sesuai yang tertera pada akun rajaongkir kita, ini berfungsi untuk

memvalidasi apakah ini benar dari pengguna rajaongkir atau tidak. Selain itu juga untuk memvalidasi akun kita termasuk yang gratis atau berbayar. Setelah semua berhasil semua data akan disimpan kedalam variabel \$response. Kemudian jika kita ingin menggunakannya kita tinggal melakukan decode pada variabel \$response. *Source code* konfigurasi rajaongkir dapat kita lihat pada gambar 4.27.

```
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => "https://api.rajaongkir.com/starter/province",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "GET",
  CURLOPT_HTTPHEADER => array(
    "key: "
  ),
));
$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);
```

Gambar 4.27 Konfigurasi rajaongkir

Fungsi `initPaymentGateway()` ini berfungsi untuk menginisialisasi pertama ke Midtrans. Kemudian karena untuk membuat transaksi menggunakan method post kita harus mengirimkan parameter informasi tentang konsumen kita sesuai nota yang telah dibuat di dcreatif unika. Untuk parameter `order_id` harus berupa kode unik sebagai identitas. Kemudian setelah semua data siap, kita mengirimkan data tersebut dengan API midtrans, dengan perintah `\Midtrans\Snap::createTransaction($params)` dengan

memasukan data konsumen tersebut ke parameter. Kemudian setelah transaksi midtrans berhasil dibuat, token payment dari midtrans akan disimpan ke database Dcreatif Unika yang nantinya akan digunakan untuk memvalidasi pembayaran oleh konsumen.

```
$this->initPaymentGateway();

$customerDetails = [
    'first_name' => $user->name,
    'last_name' => $user->name,
    'email' => $user->email,
    'phone' => $user->no_hp,
];

$params = [
    'enable_payments' => \App\Models\Payment::PAYMENT_CHANNELS,
    'transaction_details' => [
        'order_id' => $nota->id,
        'gross_amount' => $nota->total_pembayaran+$nota->ongkir,
    ],
    'customer_details' => $customerDetails,
    'expiry' => [
        'start_time' => date('Y-m-d H:i:s T'),
        'unit' => \App\Models\Payment::EXPIRY_UNIT,
        'duration' => \App\Models\Payment::EXPIRY_DURATION,
    ],
];

$snap = \Midtrans\Snap::createTransaction($params);

if ($snap->token) {
    $nota->payment_token = $snap->token;
    $nota->payment_url = $snap->redirect_url;
    $nota->save();
}
```

Gambar 4.28 Konfigurasi *Payment Gateway* Midtrans

#### 4.2.2.2. Halaman Pelapak

Halaman pemilik toko ini berfungsi untuk mengetahui informasi tentang toko tersebut, selain itu juga terdapat daftar produk yang dijual oleh toko tersebut. Pemilik toko juga

dapat mengubah dan menambah informasinya. Selain itu setiap transaksi dari konsumen akan menambah informasi jumlah saldo, dan nantinya saldo tersebut bisa diambil oleh konsumen. Tampilan halaman pemilik toko dapat kita lihat pada gambar 4.29.

The screenshot shows the admin interface for 'Dcreatif Unika'. At the top, there's a navigation bar with 'Welcome in dcreatifunika.id', 'Track Order', '081 3257 23410', and 'Logout'. Below this is a search bar and a main menu with 'Home', 'My Shop', 'List Order', 'Checkout', 'Kategori', and 'History'. The main content area is divided into two sections: 'Upload Product' and 'List Produk'.

The 'Upload Product' section contains a table with the following data:

Nama toko	Alamat toko	Kecamatan	Kota	Provinsi	Kode Pos	Saldo
Trimo Sibuk	Jl barito	Semarang Timur	Kota Semarang	Jawa Tengah	50135	Rp. 89.000,00 <a href="#">Withdraw</a>

The 'List Produk' section contains a table with the following data:

Nama Produk	Stock	Deskripsi	Berat	Kategori
 Mie Ayam Goreng	56	Mie ayam enak lezat dan bergizi, harga terjangkau dan siap kirim ke seluruh Indonesia	0.5 kg/pcs	Makanan <a href="#">Edit</a>
 T-Shirt Celine	55	bahan : spandex bahan adem dan stretch nyaman dipakai dan jatuh lingk dada : 94-110cm panjang : 47cm lingk ketiak : 36cm	0.2 kg/pcs	Fashion <a href="#">Edit</a>

At the bottom of the page, there is a footer with the text: '© 2017 Grocery Shoppy. All rights reserved | Design by W3layouts.'

Gambar 4.29 Tampilan halaman pemilik toko

Halaman pemilik toko ini berupa tabel yang berisikan informasi tentang alamat toko, provinsi, kota, kecamatan, dan kode pos. Satu akun hanya dapat mendaftarkan sekali toko. *Source code* halaman pemilik toko dapat kita lihat pada gambar 4.30.

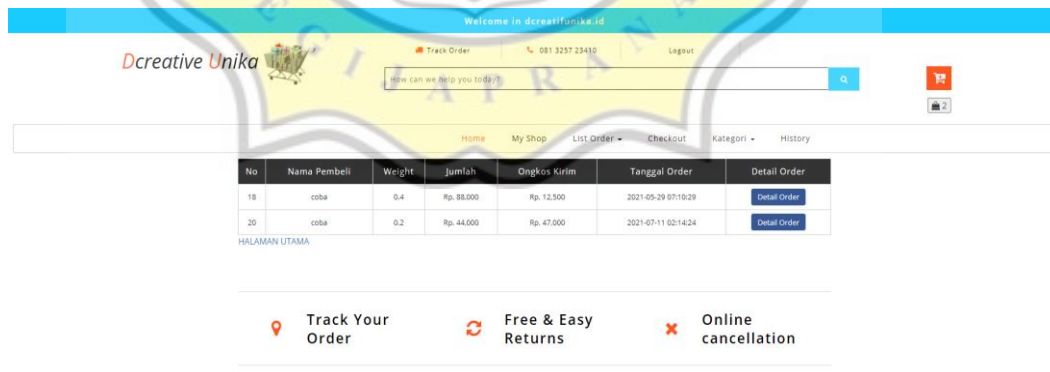
```

<tr>
  <th class="shopping_product">Nama toko</th>
  <th>Alamat toko</th>
  <th>Kecamatan</th>
  <th>Kota</th>
  <th>Provinsi</th>
  <th>Kode Pos</th>
</tr>
</thead>
<tbody>
  <tr>
    <td class="shopping_cart_price">
      {{<!-- toko -->nama_toko}}
    </td>
    <td class="shopping_cart_item">
      <center><strong>{{<!-- toko -->alamat_toko}}</strong></center>
    </td>
    <td class="shopping_cart_price">
      {{<!-- toko -->kecamatan_toko}}
    </td>
    <td class="shopping_cart_price">
      {{<!-- toko -->nama_kota}}
    </td>
    <td class="shopping_cart_price">
      {{<!-- toko -->provinsi_toko}}
    </td>
    <td class="shopping_cart_price">
      {{<!-- toko -->kode_pos}}
    </td>
    <td class="shopping_cart_price">
      Rp. {{number_format($toko->saldoo)}}.00 @if($toko->saldoo != 0) <a href="{{url('/debit')}}" class="btn btn-primary"
    </td>
  </tr>

```

Gambar 4.30 Source code halaman pemilik toko

Halaman ini untuk mengetahui daftar pesanan setiap toko. Sebelum validasi pesanan konsumen, pemilik toko wajib mengirim barang yang dipesan dan baru memasukan nomor resi pengiriman kedalam sistem (validasi pesanan). Halaman list order pemilik toko dapat kita lihat pada gambar 4.31.



Gambar 4.31 Halaman list order pemilik toko

Halaman list order pemilik toko ini berupa tabel yang terdiri dari nama pembeli, berat produk, jumlah produk yang dibeli, ongkos kirim, tanggal konsumen order, dan tombol



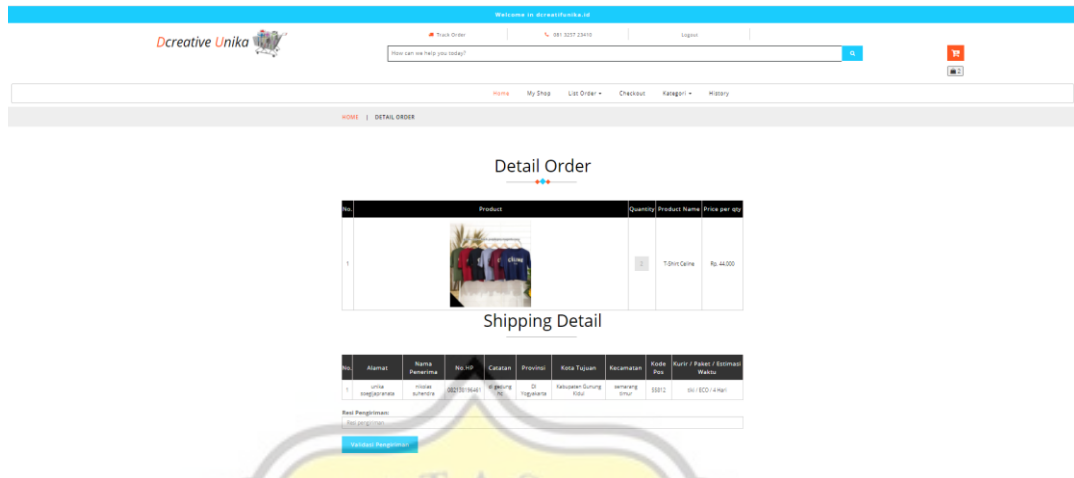
untuk melihat pesanan konsumen. *Source code* halaman *list order* pemilik toko dapat kita lihat pada gambar 4.32.

```
<thead>
<tr>
<th>No</th>
<th>Nama Pembeli</th>
<th>Weight</th>
<th>Jumlah</th>
<th>Ongkos Kirim</th>
<th>Tanggal Order</th>
<th>Detail Order </th>
</tr>
</thead>
<tbody>

@foreach($order as $order)
<tr>
<td class="shopping_cart_price">
    {{$order->id}}
</td>
<td class="shopping_cart_item">
<?php
    $namauser =\App\Models\User::where('id',$order->id_user)->first();
    >
    {{$namauser->name}}
</td>
<td class="shopping_cart_price">
    {{$order->total_weight}}
</td>
<td class="shopping_cart_price">
    Rp. {{number_format($order->total_harga)}}
</td>
<td class="shopping_cart_price">
    Rp. {{number_format($order->ongkos_kirim)}}
</td>
<td class="shopping_cart_price">
    {{$order->created_at}}
</td>
</tr>
</tbody>
```

Gambar 4.32 *Source Code* halaman list order pemilik toko

Halaman ini berfungsi agar pemilik toko mengetahui detail pesanan konsumen dan alamat konsumen, serta jumlah yang dipesan oleh konsumen. Ini memudahkan pemilik toko ketika mengirim barang sesuai dengan ekspedisi yang dipilih oleh konsumen. Detail order dan shipping detail pemilik toko dapat kita lihat pada gambar 4.33.



Gambar 4.33 Halaman detail *order* dan detail *shipping* pemilik toko

Halaman detail order dan detail shipping ini berisi informasi tentang pengiriman dan detail pesanan konsumen. Ini untuk memudahkan pemilik toko untuk mengirimkan produk beserta alamat lengkap tujuan konsumen. *Source code* detail *order* dan detail *shipping* pemilik toko dapat kita lihat pada gambar 4.34.

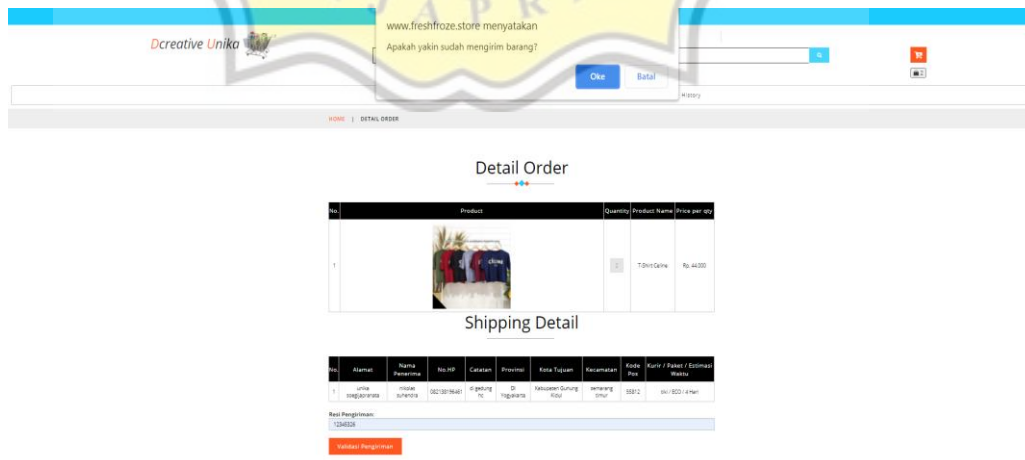
```

<thead>
<tr>
<th>No.</th>
<th>Alamat</th>
<th>Nama Penerima</th>
<th>No.HP</th>
<th>Catatan</th>
<th>Provinsi</th>
<th>Kota Tujuan</th>
<th>Kecamatan</th>
<th>Kode Pos</th>
<th>Kurir / Paket / Estimasi Waktu</th>
</tr>
</thead>
<tbody>
<tr>
<td colspan="10">
@if(!empty($orderdetail))
<?php
$no1 =1;
?>
<tr class="real">
<td class="invert">{{$no1++}}</td>
<td class="invert">{{$shipping->alamat}} </td>
<td class="invert">{{$shipping->nama_penerima}} </td>
<td class="invert">{{$shipping->no_hp}} </td>
<td class="invert">{{$shipping->catatan}} </td>
<td class="invert">{{$shipping->provinsi}} </td>
<td class="invert">{{$shipping->distrik}} {{$shipping->nama_kota}}</td>
<td class="invert">{{$shipping->kecamatan}} </td>
<td class="invert">{{$shipping->kode_pos}} </td>
<td class="invert">{{$shipping->nama_kurir}} / {{$shipping->paket}} / {{$shipping->etd}} Hari </td>
</tr>
<tr>
<td colspan="10">
@endif
</tbody>

```

Gambar 4.34 Source Code detail order dan detail shipping pemilik toko

Setelah mengirim barang pemilik toko harus memasukan nomor resi pengiriman. Kemudian akan muncul notifikasi validasi apakah yakin sudah mengirim barang. Yang nantinya akan dicek oleh konsumen jika barang sudah sampai. Validasi setelah memasukan nomor resi pengiriman dapat kita lihat pada gambar 4.35.



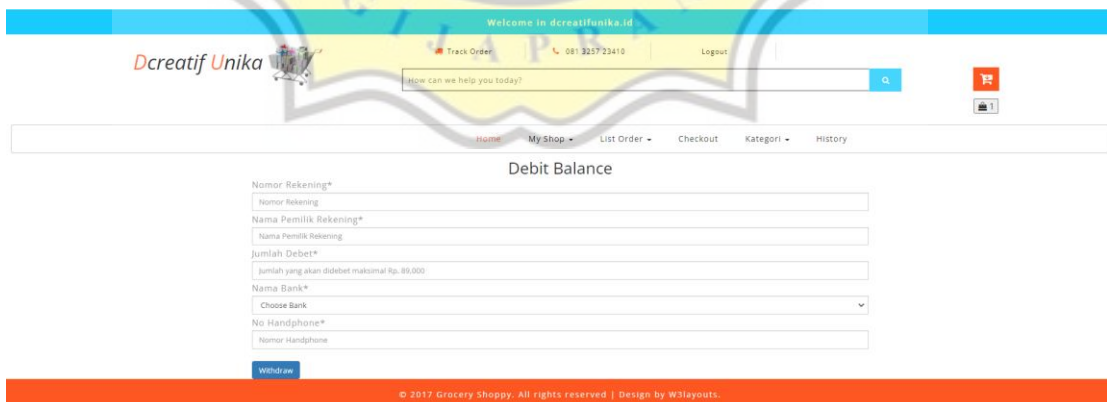
Gambar 4.35 Validasi setelah memasukan nomor resi pengiriman

Validasi ini nantinya akan dicek ke database rajaongkir untuk melakukan track nota apakah pemilik toko benar sudah mengirimkan barang atau belum. Namun, untuk saat ini belum bisa karena terkendala dengan akun yang berbayar dari pihak rajaongkir. *Source code* untuk validasi nomor resi pengiriman dapat kita lihat pada gambar 4.36.

```
<form action="{{url('validasikirim')}}{{order->id}}" method="post" class="credity-card-form agileinfo_form">
  {{csrf_field()}}
<br/> <label> Resi Pengiriman: </label> <br/><input class="form-control" type="text" name="resipengiriman" placeholder="Resi pengiriman"
<button class="submit check_out" type="submit" onclick="return confirm('Apakah yakin sudah mengirim barang?')>Validasi Pengiriman
</button>
</form>
```

Gambar 4.36 *Source Code* validasi nomor resi pengiriman

Halaman ini berfungsi untuk pengajuan debit saldo oleh pemilik toko, semua pesanan yang dilakukan oleh konsumen akan ditampung di rekening Midtrans Dcreatif Unika. Kemudian setelah semua pesanan dianggap sukses baru disalurkan kepada pemilik toko. Halaman untuk melakukan debit saldo pemilik toko dapat dilihat pada gambar 4.37.



Gambar 4.37 Halaman untuk melakukan debit saldo pemilik toko

Formulir untuk melakukan debit saldo terdiri dari formulir input yang nanti akan diisi oleh pemilik toko untuk pengajuan debit saldo, seperti identitas bank, nomor rekening, nama penerima, nomor hp, nama pemilik rekening, dan *list dropdown* bank. *Source code* debit saldo pemilik toko dapat kita lihat pada gambar 4.38.

```

<div class="container">
  <center>Debit Balance</center>
  <form method="POST" action="{(route('postdebitsaldo'))}">
    {{csrf_field()}}
    <p>Nomor Rekening<input type="text" placeholder="Nomor Rekening" class="form-control" name="norekening" required/></p>
    <p>Nama Pemilik Rekening<input type="text" placeholder="Nama Pemilik Rekening" class="form-control" name="namapemilik" required/></p>
    <p>Jumlah Debit<input type="number" max="{(toko->saldo)}" name="jumlahdebit" placeholder="Jumlah yang akan didebet maksim"></p>
    <p>Nama Bank<select class="form-control" name="namabank" placeholder="Please Choose Bank">
      <option value=""> Choose Bank</option>
      <option value="Bank Mandiri">Bank Mandiri</option>
      <option value="Bank Bukopin">Bank Bukopin</option>
      <option value="Bank Danamon">Bank Danamon</option>
      <option value="Bank Mega">Bank Mega</option>
      <option value="Bank CIMB Niaga">Bank CIMB Niaga</option>
      <option value="Bank Permata">Bank Permata</option>
      <option value="Bank Sinarmas">Bank Sinarmas</option>
      <option value="Bank QNB">Bank QNB</option>
      <option value="Bank Lippo">Bank Lippo</option>
      <option value="Bank UOB">Bank UOB</option>
      <option value="Panin Bank">Panin Bank</option>
      <option value="Citibank">Citibank</option>
      <option value="Bank ANZ">Bank ANZ</option>
      <option value="Bank Commonwealth">Bank Commonwealth</option>
      <option value="Bank Maybank">Bank Maybank</option>
      <option value="Bank Maspion">Bank Maspion</option>
      <option value="Bank J Trust">Bank J Trust</option>
      <option value="Bank QNB">Bank QNB</option>
      <option value="Bank KEB Hana">Bank KEB Hana</option>
      <option value="Bank Artha Graha">Bank Artha Graha</option>
      <option value="Bank OCBC NISP">Bank OCBC NISP</option>
      <option value="Bank MNC">Bank MNC</option>
      <option value="Bank DBS">Bank DBS</option>
    </select></p>
  </form>
</div>

```

Gambar 4.38 *Source code* debit saldo pemilik toko

### Halaman Admin

Task scheduler yang berfungsi untuk membuat pesanan menjadi kadaluarsa dan tidak dapat dilanjutkan lagi. Pesanan dianggap kadaluarsa jika batas pembayaran melewati batas waktu yang ditentukan. Jadi, task scheduler ini akan dieksekusi setiap 5 menit sekali. Proses di balik layar expired order ini dapat kita lihat pada gambar 4.39.

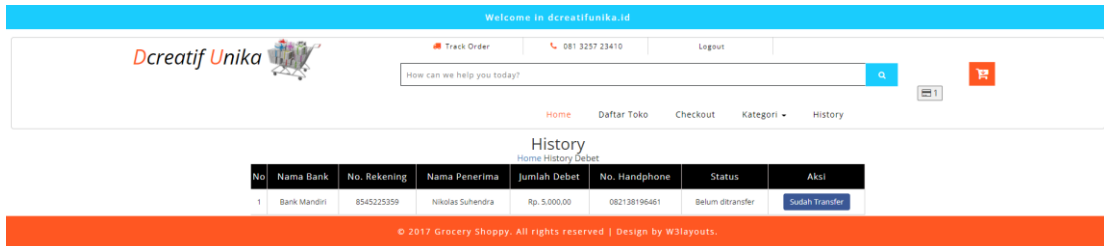
```

public function handle()
{
    //return 0;
    $tanggal=Carbon::now();
    $tglorder=$tanggal->timezone('Asia/Jakarta');
    $nota=nota::where('payment_due','<',$tglorder)->where('payment_status'
    ,"unpaid")->get();
    foreach($nota as $nota){
        $nota->payment_status="expired";
        $nota->proses=6;
        $orderdetail= OrderDetail::where('id_nota',$nota->id)->get();
        foreach($orderdetail as $orderdetail){
            $produk=Produsen::where('id',$orderdetail->id_produk)->first();
            $produk->stock=$produk->stock+$orderdetail->jumlah;
            $produk->update();
            $nota->update();
            $orderdetail->expired=1;
            $orderdetail->update();
        }
        echo "Operation done!";
    }
}

```

Gambar 4.39 Proses di balik layar *expired order*

Halaman riwayat penarikan saldo ini berguna untuk mengetahui semua pemilik toko yang ingin menarik saldo di Dcreatif Unika. Setelah admin mengirimkan uang sesuai dengan jumlah yang didebet oleh pemilik toko, admin harus memvalidasi tombol sudah transfer, agar saldo pemilik toko terpotong. Halaman riwayat penarikan saldo dapat dilihat pada gambar 4.40.



Gambar 4.40 Halaman riwayat penarikan saldo

Validasi tombol sudah transfer ini menghitung saldo tiap toko kemudian akan dikurangkan dengan jumlah yang sudah didebet. *Source code* tombol sudah transfer dapat kita lihat pada gambar 4.41.

```

public function sudahtransfer($id){
    $debet1 = debit::where('id',$id)->where('status',0)->first();
    if(!empty($debet1)){
        $user = Toko::where('id_pemilik',$debet1->id_user)->first();
        $user->saldo = $user->saldo-$debet1->jumlah;
        $debet1->status = 1;
        $user->update();
        $debet1->update();
    }
    $debet = debit::where('status',0)->get();
    return view('history.historydebit',compact('debit'));
}

```

Gambar 4.41 *Source code* tombol sudah transfer

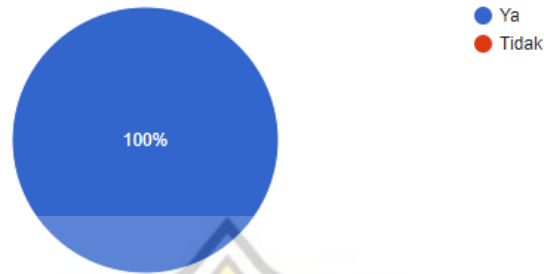
### 4.3. Pengujian dan Analisis

#### 4.3.1. Pre-test

Dari gambar 4.42 diagram lingkaran banyaknya responden yang pernah menggunakan E-Marketplace, kita ketahui bahwa seluruh responden pernah menggunakan E-Marketplace. Hal ini dibuktikan dengan diagram diatas dengan persentase 100%.

Apakah kamu pernah menggunakan E-Marketplace? (contoh: tokopedia, shopee, lazada)

30 responses

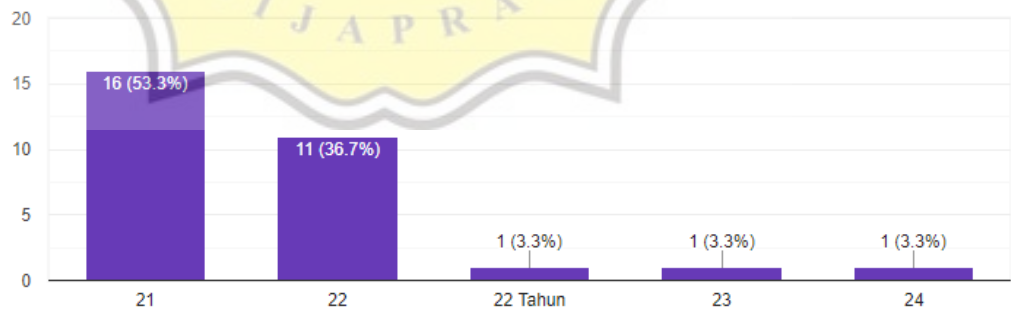


Gambar 4.42 Diagram Lingkaran banyaknya responden yang pernah menggunakan E-Marketplace

Rentang umur responden antara 21-24 tahun, dan responden yang paling banyak umur 21 tahun yaitu dengan jumlah 16 dengan persentase 53.3% dari keseluruhan. Kita dapat melihat diagram batang gambar 4.43.

Berapa umur kamu? (dalam tahun contoh: 19,20,21)

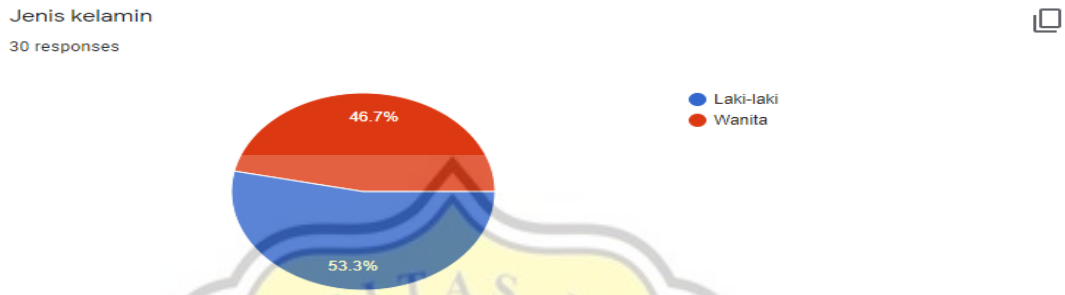
30 responses



Gambar 4.43 Diagram batang tentang umur responden

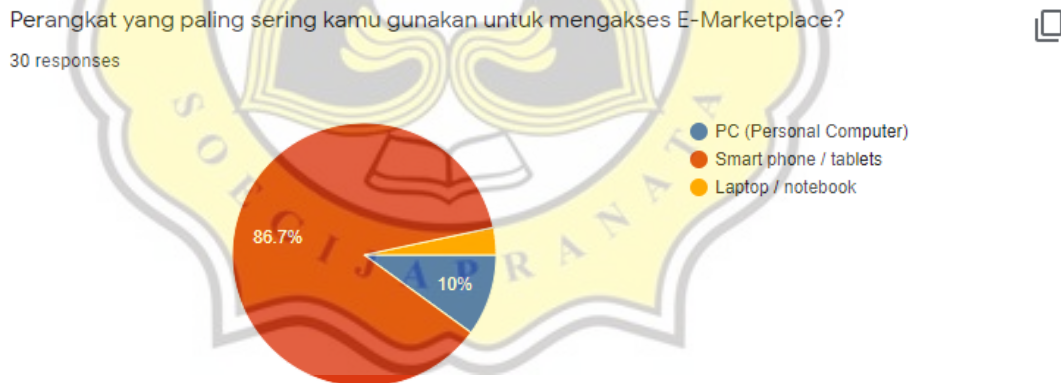


Dari gambar diagram lingkaran 4.44, kita mengetahui bahwa jumlah responden pria lebih banyak dengan persentase 53.3% dari pada wanita dengan persentase 46.7%.



Gambar 4.44 Diagram lingkaran perbandingan jenis kelamin responden

Dari gambar diagram lingkaran 4.45 kita mengetahui bahwa paling banyak responden mengakses E-Marketplace menggunakan Handphone/Tablets dengan jumlah persentase 86.7%.

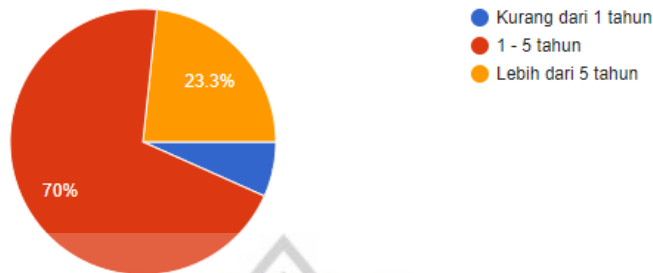


Gambar 4.45 Diagram lingkaran tentang perangkat yang paling sering digunakan untuk mengakses E-Marketplace

Dari gambar diagram lingkaran 4.46 kita mengetahui bahwa responden menggunakan E-Marketplace paling banyak dengan rentang kurun waktu 1-5 tahun.

Sudah berapa lama kamu menggunakan E-Marketplace?

30 responses

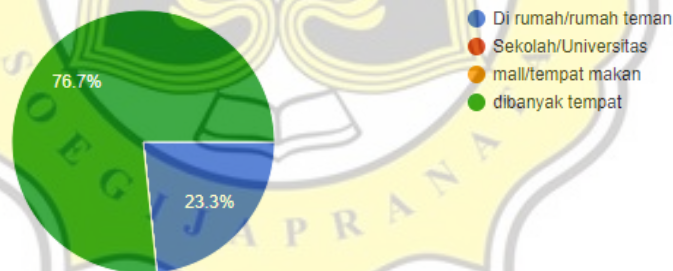


Gambar 4.46 Diagram lingkaran lamanya menggunakan E-Marketplace

Dari gambar diagram lingkaran 4.47 kita mengetahui bahwa responden paling banyak mengakses E-Marketplace di banyak tempat dengan persentase 76.7%.

Dimana anda biasa menggunakan E-Marketplace?

30 responses

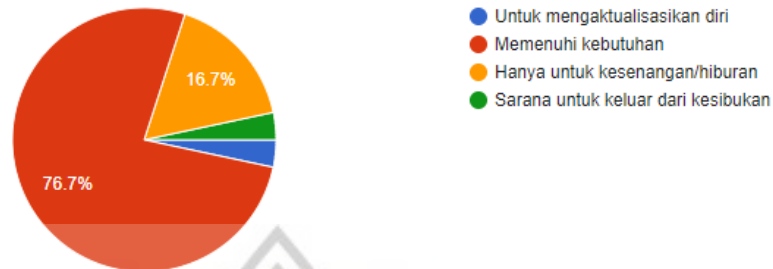


Gambar 4.47 Diagram lingkaran lokasi saat menggunakan E-Marketplace

Dari gambar diagram lingkaran 4.48 kita mengetahui bahwa alasan utama responden menggunakan E-Marketplace adalah untuk memenuhi kebutuhan dengan persentase 76.7%.

Apa alasan utama anda menggunakan E-Marketplace?

30 responses

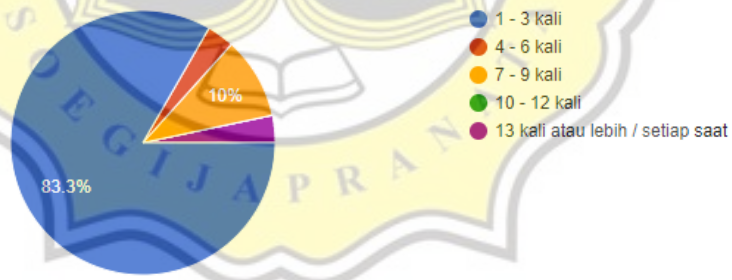


Gambar 4.48 Diagram lingkaran alasan menggunakan E-Marketplace

Dari gambar diagram lingkaran 4.49 kita mengetahui bahwa dalam sehari responden mengakses E-Marketplace kebanyakan 1 hingga 3 kali saja.

Rata-rata berapa kali dalam sehari menggunakan E-Marketplace?

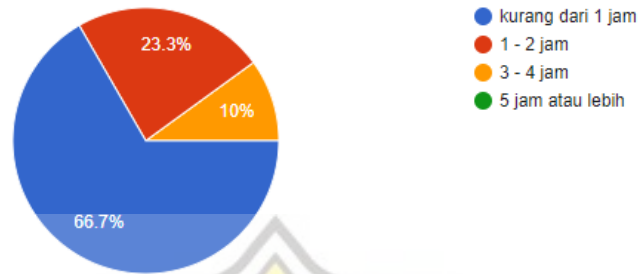
30 responses



Gambar 4.49 Diagram lingkaran jumlah mengakses E-Marketplace dalam sehari

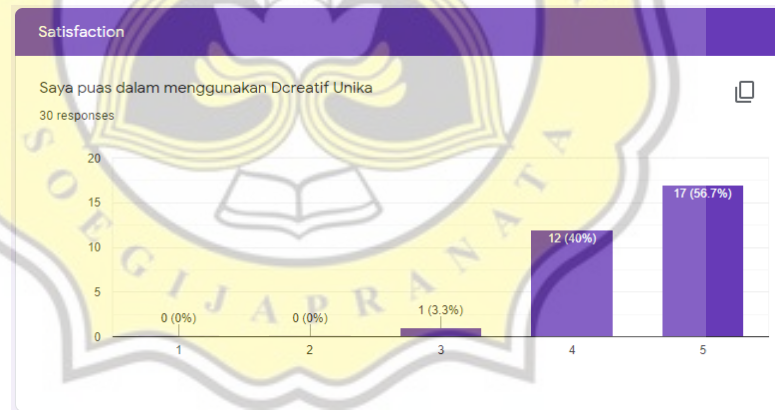
Dari gambar diagram lingkaran 4.50 kita mengetahui bahwa responden tiap kali mengakses E-Marketplace kebanyakan kurang dari 1 jam dengan persentase 66.7%.

Berapa jam waktu yang anda habiskan setiap kali anda menggunakan E-Marketplace?  
30 responses



Gambar 4.50 Diagram lingkaran lama waktu untuk sekali mengakses E-Marketplace

Dari gambar diagram batang 4.51 kita mengetahui bahwa 1 orang (3,3%) menjawab netral, 12 (40%) orang setuju dan 17 orang (56,7%) sangat setuju, puas menggunakan Dcreatif Unika.



Gambar 4.51 Diagram lingkaran lama waktu untuk sekali mengakses E-Marketplace

#### 4.3.2. Uji Validitas Dcreatif Unika

Dari gambar uji validitas pertama 4.52, terdapat 2 variabel yang tidak mengelompok yaitu FC4 dan BI1 sehingga perlu dilakukan uji validitas kedua dengan membuang kedua variabel tersebut.

**Component Matrix<sup>a</sup>**

	Component						
	1	2	3	4	5	6	7
PE1	.453	.379	.525	.263	.091	.120	-.396
PE2	.521	.263	.216	-.203	-.079	.334	-.045
PE3	.553	.286	.208	.466	.032	-.182	-.154
PE4	.545	-.258	.052	.482	.416	-.114	-.129
EE1	.480	.414	-.255	-.380	.238	-.207	-.036
EE2	.472	.606	-.244	-.251	.141	-.151	-.235
EE3	.689	.400	-.081	-.317	.090	.036	-.014
SI1	.519	-.250	.688	-.197	.131	.169	.056
SI2	.472	-.263	.679	-.205	-.069	-.101	.216
SI3	.472	-.389	.554	-.451	.053	.038	-.051
FC1	.350	.497	.161	.426	-.112	-.125	.354
FC2	.307	.792	-.026	.135	.062	-.172	.053
FC3	.468	.510	-.186	-.161	.423	.303	.160
FC4	.454	.239	-.060	.038	-.074	.128	.721
H1	.674	.189	-.323	-.053	-.064	.413	-.196
H2	.492	-.532	-.159	.060	-.055	.462	-.051
H3	.587	-.493	-.419	.119	-.021	.286	.070
H4	.609	.189	-.027	-.047	-.205	.071	.027
BI1	.378	-.425	.082	-.148	.445	-.390	.084
BI2	.677	-.368	-.185	.184	.288	-.216	.111
BI3	.527	-.555	-.314	.350	.221	.055	.004
HM1	.674	-.187	-.256	-.208	-.466	-.267	-.113
HM2	.739	-.121	-.127	-.072	-.522	-.301	-.015
HM3	.704	-.359	-.207	-.177	-.074	-.234	-.050
S1	.594	.111	.263	.493	-.357	.056	-.067

Extraction Method: Principal Component Analysis.

a. 7 components extracted.

Gambar 4.52 Tabel uji validitas pertama

Setelah melakukan uji validitas kedua ternyata semua variabel sudah mengelompok dan nilainya melebihi 0.4 sehingga tidak diperlukan untuk pengujian yang ketiga. Kita dapat melihat pada gambar 4.53.

**Component Matrix<sup>a</sup>**

	Component					
	1	2	3	4	5	6
PE1	.474	.393	.514	.266	.129	-.132
PE2	.519	.258	.212	-.215	.075	-.472
PE3	.558	.274	.208	.472	-.072	.177
PE4	.540	-.241	.052	.513	.295	.281
EE1	.484	.424	-.279	-.345	.137	.409
EE2	.489	.622	-.274	-.215	.048	.158
EE3	.684	.407	-.099	-.298	.089	-.032
SI1	.513	-.235	.691	-.212	.232	.008
SI2	.462	-.257	.686	-.228	-.065	.214
SI3	.470	-.362	.554	-.464	.114	.100
FC1	.352	.477	.160	.426	-.189	.043
FC2	.314	.789	-.046	.162	-.059	.122
FC3	.467	.500	-.202	-.133	.521	-.012
H1	.688	.153	-.320	-.072	.154	-.394
H2	.497	-.584	-.129	.012	.231	-.249
H3	.584	-.527	-.398	.097	.131	-.234
H4	.617	.179	-.030	-.060	-.155	-.229
BI2	.658	-.354	-.185	.214	.132	.297
BI3	.518	-.563	-.300	.358	.213	.048
HM1	.675	-.206	-.249	-.234	-.522	.010
HM2	.743	-.150	-.116	-.106	-.585	.047
HM3	.702	-.366	-.203	-.181	-.147	.246
S1	.611	.073	.279	.453	-.294	-.258

Extraction Method: Principal Component Analysis.

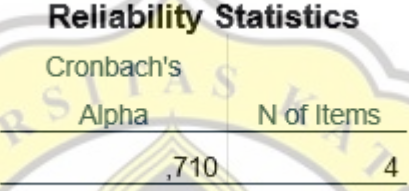
a. 6 components extracted.

Gambar 4.53 Tabel uji validitas kedua

### 4.3.3. Uji Reliabilitas Dcreatif Unika

#### 4.3.3.1. Variabel Performance Expectancy

Dari gambar 4.54 dibawah nilai Cronbach's Alpha 0.710 berarti variabel Performance Expectancy tergolong Acceptable

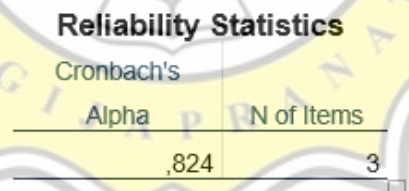


Reliability Statistics	
Cronbach's Alpha	N of Items
.710	4

Gambar 4.54 Tabel uji reliabilitas variabel Performance Expectancy

#### 4.3.3.2. Variabel Effort Expectancy

Dari gambar 4.55 dibawah nilai Cronbach's Alpha 0.824 berarti variabel Effort Expectancy tergolong good



Reliability Statistics	
Cronbach's Alpha	N of Items
.824	3

Gambar 4.55 Tabel uji reliabilitas variabel Effort Expectancy

#### 4.3.3.3. Variabel Social Influence

Dari gambar 4.56 dibawah nilai Cronbach's Alpha 0.916 berarti variabel Social Influence tergolong excellent

<b>Reliability Statistics</b>	
Cronbach's	
Alpha	N of Items
,916	3

Gambar 4.56 Tabel uji reliabilitas variabel Social Influence

#### 4.3.3.4.. Variabel Facilitating Conditions

Dari gambar 4.57 dibawah nilai Cronbach's Alpha 0.712 berarti variabel Facilitating Conditions tergolong Acceptable

<b>Reliability Statistics</b>	
Cronbach's	
Alpha	N of Items
,712	3

Gambar 4.57 Tabel uji reliabilitas variabel Facilitating Conditions

#### 4.3.3.5. Variabel Habit

Dari gambar 4.58 dibawah nilai Cronbach's Alpha 0.758 berarti variabel Habit tergolong Acceptable

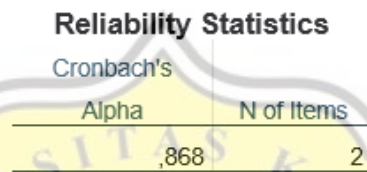
<b>Reliability Statistics</b>	
Cronbach's	
Alpha	N of Items
,758	4



Gambar 4.58 Tabel uji reliabilitas variabel Habit

#### 4.3.3.6. Variabel Behavioral Intention

Dari gambar 4.59 dibawah nilai Cronbach's Alpha 0.824 berarti variabel Behavioral Intention tergolong good

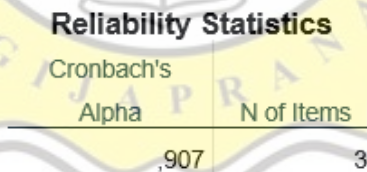


Reliability Statistics	
Cronbach's Alpha	N of Items
.868	2

Gambar 4.59 Tabel uji reliabilitas variabel Behavioral Intention

#### 4.3.3.7. Variabel Hedonic Motivation

Dari gambar 4.60 dibawah nilai Cronbach's Alpha 0.907 berarti variabel Hedonic Motivation tergolong Excellent



Reliability Statistics	
Cronbach's Alpha	N of Items
.907	3

Gambar 4.60 Tabel uji reliabilitas variabel Hedonic Motivation

#### 4.3.4. Uji Korelasi Dcreatif Unika

Setelah melakukan uji korelasi, kita dapat mengetahui variabel-variabel yang berkorelasi dengan melihat nilai uji significant-nya (2-

tailed) < 0.05 pada hasil uji korelasi. Kita dapat melihat hasil uji korelasi pada gambar 4.61.

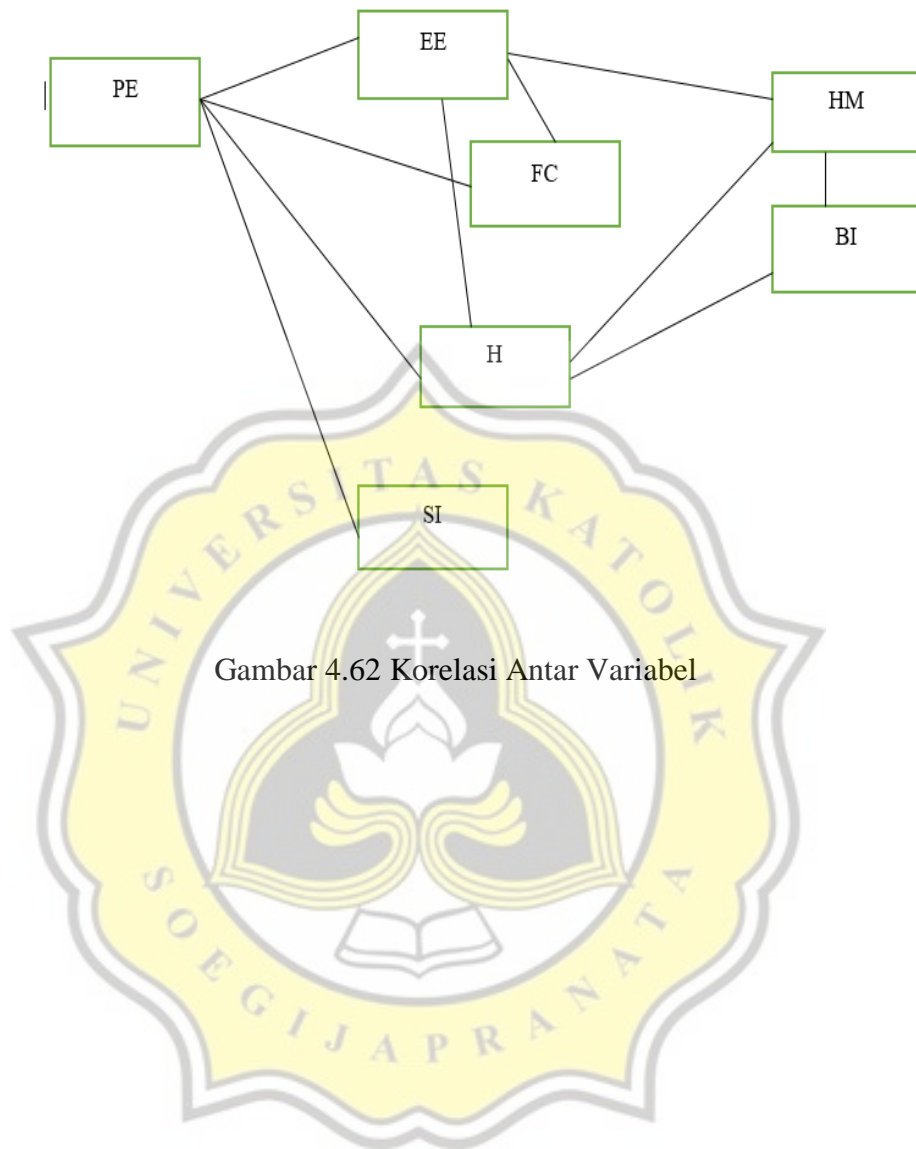
		Correlations						
		RPE	REE	RSI	RFC	RH	RBI	RHM
RPE	Pearson Correlation	1	,386*	,373*	,480**	,377*	,323	,342
	Siq. (2-tailed)		,035	,042	,007	,040	,082	,064
	N	30	30	30	30	30	30	30
REE	Pearson Correlation	,386*	1	,139	,611**	,369*	,147	,410*
	Siq. (2-tailed)	,035		,463	,000	,045	,437	,024
	N	30	30	30	30	30	30	30
RSI	Pearson Correlation	,373*	,139	1	-,010	,270	,256	,344
	Siq. (2-tailed)	,042	,463		,959	,149	,172	,063
	N	30	30	30	30	30	30	30
RFC	Pearson Correlation	,480**	,611**	-,010	1	,170	,048	,105
	Siq. (2-tailed)	,007	,000	,959		,368	,799	,582
	N	30	30	30	30	30	30	30
RH	Pearson Correlation	,377*	,369*	,270	,170	1	,605**	,598**
	Siq. (2-tailed)	,040	,045	,149	,368		,000	,000
	N	30	30	30	30	30	30	30
RHM	Pearson Correlation	,342	,410*	,344	,105	,598**	,512**	1
	Siq. (2-tailed)	,064	,024	,063	,582	,000	,004	
	N	30	30	30	30	30	30	30
RBI	Pearson Correlation	,323	,147	,256	,048	,605**	1	,512**
	Siq. (2-tailed)	,082	,437	,172	,799	,000		,004
	N	30	30	30	30	30	30	30

\*. Correlation is significant at the 0.05 level (2-tailed).

\*\*. Correlation is significant at the 0.01 level (2-tailed).

Gambar 4.61 Hasil uji korelasi

Dari uji korelasi kita dapat menyimpulkan bahwa hubungan variabel PE berkorelasi dengan EE, FC, H, dan SI hal ini dibuktikan dengan nilai significant variabel diatas kurang 0.05. Namun PE tidak berkorelasi dengan BI dan HM,menunjukkan bahwa persepsi kebergunaan Dcreatif belum menjamin orang untuk menyukai dan bersedia terus menggunakannya. Kita dapat melihat hubungan antar variabel pada gambar 4.62.



Gambar 4.62 Korelasi Antar Variabel