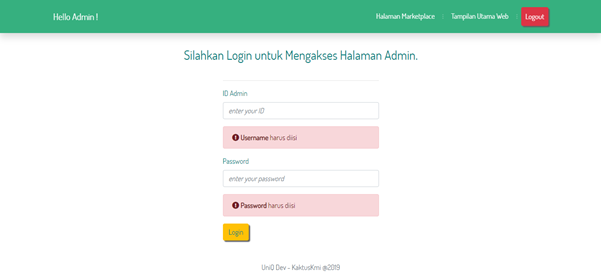
* + - 1. Jadwal Programmer User Interface (Intan Destiyanti)

1. Minggu – selasa, 5 Mei 2019 - 7 Mei 2019

Kegiatan : (Membuat validasi halaman login admin)

Dibagian form login jika inputan tidak diisi maka akan muncul alert yaitu password dan username harus di isi

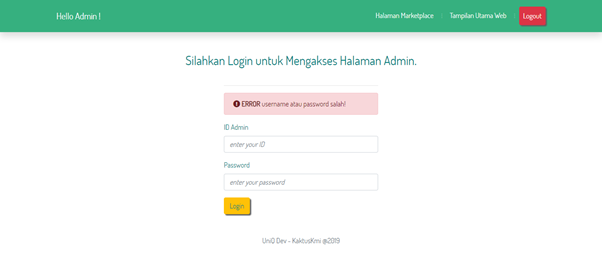


Gambar x.x validasi login

|  |
| --- |
| //set form validation  $this->form\_validation->set\_rules('username','Username', 'required');  $this->form\_validation->set\_rules('password','Password', 'required');  //set message form validation  $this->form\_validation->set\_message('required','<div class="alert alert-danger" style="margin-top: 3px"><div class="header"><b><i class="fa fa-exclamation-circle"></i> {field}</b> harus diisi</div></div>');  //set form validation  $this->form\_validation->set\_rules('username','Username', 'required');  $this->form\_validation->set\_rules('password','Password', 'required');  //set message form validation  $this->form\_validation->set\_message('required','<div class="alert alert-danger" style="margin-top: 3px"><div class="header"><b><i class="fa fa-exclamation-circle"></i> {field}</b> harus diisi</div></div>'); |

Source code x.x validasi form login

Kodingan pada source code x.x berisi rule untuk validasi pada bagian username dan password yaitu berarti username dan password tidak boleh kosong, dan juga pesan yang akan muncul jika username dan password tersebut kosong.



Gambar x.x validasi sesuai database

Pada tampilan Di atas yaitu jika username atau password tidak sesuai dengan database maka akan muncul alert “ERROR username atau password salah!”.

Berikut contoh script pada model dengan nama file Login\_Model.php berisi query untuk mengambil semua data sesuai dengan parameter tertera.

|  |
| --- |
| <?php  class Login\_Model extends CI\_Model  {  function check\_login($table, $field1, $field2)  {  $this->db->select('\*');  $this->db->from($table);  $this->db->where($field1);  $this->db->where($field2);  $this->db->limit(1);  $query = $this->db->get();  if ($query->num\_rows() == 0) {  return FALSE;  } else {  return $query->result();  }  }  }  ?> |

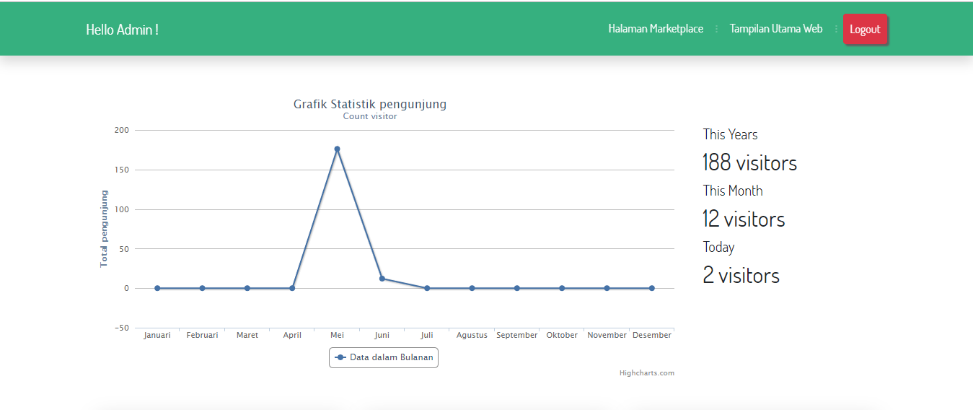
Source code x.x query check login

Berikut contoh script pada bagian controller dengan file bernama Login.php berisi perintah tentang membuat validasi pada form login dan mengecek username dan password sesuai dengan database.

|  |
| --- |
| <?php  class Login extends CI\_Controller  {     function \_\_construct(){         parent::\_\_construct();         $this->load->model('Login\_Model');         $this->load->library('form\_validation');       public function index(){         $judul['judul'] = 'Halaman Admin';         $this->load->view('templates/header\_admin', $judul);         $this->load->view('adminpage/loginpage');         $this->load->view('templates/footer');       public function aksi(){         //set form validation         $this->form\_validation->set\_rules('username', 'Username', 'required');         $this->form\_validation->set\_rules('password', 'Password', 'required');         //set message form validation         $this->form\_validation->set\_message('required',     '<div class="alert alert-danger" style="margin-top: 3px"><div class="header"><b><i class="fa fa-exclamation-circle"></i> {field}</b> harus diisi</div></div>');         if ($this->form\_validation->run() == TRUE) {             $username = $this->input->post('username');             $password = $this->input->post('password');             $where = array(                 'username' => $username,                 'password' => $password);  //checking data via model        $checking = $this->Login\_Model->check\_login('penjual',        array('username' => $username),array('password' => $password));            if ($checking > 0) {        $session\_data = array(           'username' => $username,           'password' => $password,           'status' => "Login");                                 $this->session->set\_userdata($session\_data);            redirect("Adminpage/Home/");        else{       $data['error'] = '<div class="alert alert-danger" style="margin-top: 3px"><div class="header"><b><i class="fa fa-exclamation-circle"></i> ERROR</b> username atau password salah!</div></div>';              $judul['judul'] = 'Halaman Admin';            $this->load->view('templates/header\_admin', $judul);            $this->load->view('adminpage/loginpage',$data);            $this->load->view('templates/footer');        }}else{            $judul['judul'] = 'Halaman Admin';            $this->load->view('templates/header\_admin', $judul);            $this->load->view('adminpage/loginpage');            $this->load->view('templates/footer');        }}         public function logout()              $this->session->sess\_destroy();            redirect(base\_url('Login'));      ?> |

Source code x.x controller login

1. Rabu – Jumat, 8 - 10 Mei 2019 dan Kamis, 13 Mei 2019

Kegiatan : Membuat statistik pengunjung

Gambar x.x statistik pengunjung

Di halaman awal pada bagian admin disini terdapat statistik untuk menghitung jumlah pengunjung yang mengunjungi web penjualan kaktuskmi pada bagian grafik menampilkan statistik pengunjung perbulannya kemudian di samping grafik dapat menampilkan jumlah pengunjung pertahun, perbulan dan perhari. Kemudian grafiknya menggunakan highchart yang berbasic javascript.

Berikut contoh script pada bagian views dengan nama file index.php berisi perintah untuk menampilkan grafik dengan format perbulan dan perintah untuk menampilkan jumlah pengunjung pertahun, perbulan dan perhari.

|  |
| --- |
| <section>      <div class="container my-5">          <!-- Statistik -->          <div class="row">              <div class="col-sm-9">                      <div id="chart"></div>              </div>              <div class="col mt-5">              <?php                  foreach($pertahun as $tahun){                  echo'                      <h5> This Years</h5>                      <h2>'.$tahun->jumlah\_tahun.' visitors</h2>                  ;                    foreach($perbulan as $bulan){                      echo'                      <h5> This Month</h5>                      <h2>'.$bulan->jumlah\_bulan.' visitors</h2>                      ;                    foreach($perhari as $harian){                      echo'                          <h5> Today </h5>                          <h2>'.$harian->hari.' visitors</h2>                      ;                ?>              </div>          </div>  <script type="text/javascript">  jQuery(function(){  new Highcharts.Chart({  chart: {  renderTo: 'chart',  type: 'line',  },  title: {  text: 'Grafik Statistik pengunjung',  x: -20  },  subtitle: {  text: 'Count visitor',  x: -20  },  xAxis: {  categories: ['Januari','Februari','Maret','April','Mei','Juni','Juli','Agustus','September','Oktober','November','Desember']  },  yAxis: {  title: {  text: 'Total pengunjung'  }  },  series: [{  name: 'Data dalam Bulanan',  data: <?php echo json\_encode($grafik); ?>  }]  });  });  </script> |

Source code x.x grafik statistik

Berikut contoh script model dengan file bernama Counter\_Model.php berisi perintah untuk mengambil ip pengunjung kemudian dihitung setiap ip itu yang mengunjungi web penjualan kaktuskmi yang nntinya akan tersimpan kedalam database sebagai data membuat grafik statistik pengunjung.

|  |
| --- |
| <?php  class Counter\_Model extends CI\_Model  {  private $\_table="tbcounter";  function Model\_Counter()  {  function ambil\_ip\_pengunjung() {    if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])) {        $ip = $\_SERVER['HTTP\_CLIENT\_IP'];            } elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {        $ip = $\_SERVER['HTTP\_X\_FORWARDED\_FOR'];    } else {        $ip = $\_SERVER['REMOTE\_ADDR'];          return $ip;    }    $counter = array(    'ip' => ambil\_ip\_pengunjung(),    'date' => date("Y-m-d"),    'hits' => '1'    );       $this->db->where('ip', $counter['ip']);     $this->db->where('date', $counter['date']);     $query = $this->db->get($this->\_table, FALSE);     $cek = $query->num\_rows();     if($cek > 0){       $hits = $query->row()->hits;       $hits++;         $data = array(             'ip' => $counter['ip'],             'date' => $counter['date'],             'hits' => $hits       );       $this->db->where('ip', $counter['ip']);       $this->db->where('date', $counter['date']);       $this->db->update($this->\_table,$data);       else{       $data = array(             'ip' => $counter['ip'],             'date' => $counter['date'],              'hits' => '1'       );       $this->db->insert($this->\_table, $data);      }  ?> |

Source code x.x insert data pengunjung

Berikut contoh script pada bagian model dengan file bernama Model\_grafik.php berisi queri pada fungsi pertama yaitu statistik\_pengunjung\_perbulan() menampilkan jumlah pengunjung perbulan sesuai dengan bulan sekarang dan tahun sekarang yang nantinya sebagai data pada grafik pengunjung sedangkan untuk fungsi lainnya yaitu query untuk menghitung perhari, perbulan dan pertahun.

|  |
| --- |
| <?php  class Model\_grafik extends CI\_Model {     function statistik\_pengunjung\_perbulan(){     $sql= $this->db->query("      select      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=1) AND (YEAR(date)=YEAR(NOW())))),0) AS `Januari`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=2) AND (YEAR(date)=YEAR(NOW())))),0) AS `Februari`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=3) AND (YEAR(date)=YEAR(NOW())))),0) AS `Maret`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=4) AND (YEAR(date)=YEAR(NOW())))),0) AS `April`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=5) AND (YEAR(date)=YEAR(NOW())))),0) AS `Mei`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=6) AND (YEAR(date)=YEAR(NOW())))),0) AS `Juni`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=7) AND (YEAR(date)=YEAR(NOW())))),0) AS `Juli`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=8) AND (YEAR(date)=YEAR(NOW())))),0) AS `Agustus`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=9) AND (YEAR(date)=YEAR(NOW())))),0) AS `September`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=10) AND (YEAR(date)=YEAR(NOW())))),0) AS `Oktober`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=11) AND (YEAR(date)=YEAR(NOW())))),0) AS `November`,      ifnull((SELECT SUM(hits) FROM (tbcounter)WHERE((Month(date)=12) AND (YEAR(date)=YEAR(NOW())))),0) AS `Desember`      from tbcounter GROUP BY YEAR(date)");      return $sql;         function perhari(){         $this->db->select("\*");         $this->db->select('SUM(hits) as hari');         $this->db->from("tbcounter");         $this->db->where('date = DATE(NOW())');         $this->db->group\_by("DATE(date)");         return $this->db->get()->result();       function perbulan(){         $this->db->select("MONTH(date) as bulan");         $this->db->select('SUM(hits) as jumlah\_bulan');         $this->db->from("tbcounter");  $this->db->where('MONTH(date) = MONTH(NOW()) AND YEAR(date) =  YEAR(NOW())');         $this->db->group\_by("MONTH(date)");         return $this->db->get()->result();       function pertahun(){         $this->db->select("YEAR(date) as tahun");         $this->db->select('SUM(hits) as jumlah\_tahun');         $this->db->from("tbcounter");         $this->db->where('YEAR(date) = YEAR(NOW())');         $this->db->group\_by("YEAR(date)");         return $this->db->get()->result();  }  ?> |

Source code x.x menampilkan data statistik pengunjung

Berikut contoh script pada bagian controller yang filenya bernama Home.php yang berisi perintah untuk memanggil file pada model berisi query yang nantinya data tersebut dipakai pada grafik yang sudah dibuat, pada bagian fungsi construct terdapat session status login dimana berfungsi saat user memasukin halaman home pertama kali maka akan menuju halaman login terlebih dahulu.

|  |
| --- |
| <?php  Class Home extends CI\_Controller  {     function \_\_construct(){         parent::\_\_construct();         if($this->session->userdata('status') != "Login"){             redirect(base\_url("Login"));           $this->load->model("Model\_grafik");       Public function index()           $judul['judul'] = 'Halaman Admin';      foreach($this->Model\_grafik->statistik\_pengunjung\_perbulan()->result\_array() as $row)            $data['grafik'][0]=(float)$row['Januari'];          $data['grafik'][1]=(float)$row['Februari'];          $data['grafik'][2]=(float)$row['Maret'];          $data['grafik'][3]=(float)$row['April'];          $data['grafik'][4]=(float)$row['Mei'];          $data['grafik'][5]=(float)$row['Juni'];          $data['grafik'][6]=(float)$row['Juli'];          $data['grafik'][7]=(float)$row['Agustus'];          $data['grafik'][8]=(float)$row['September'];          $data['grafik'][9]=(float)$row['Oktober'];          $data['grafik'][10]=(float)$row['November'];          $data['grafik'][11]=(float)$row['Desember'];           $data["perbulan"] = $this->Model\_grafik->perbulan();         $data["pertahun"] = $this->Model\_grafik->pertahun();         $data["perhari"] = $this->Model\_grafik->perhari();         $this->load->view('templates/header\_admin', $judul);         $this->load->view('adminpage/index',$data);         $this->load->view('templates/footer');  }  ?> |

Source code x.x controller statistik pengunjung

1. Selasa – Jumat, 14 - 17 Mei 2019

Kegiatan : Membuat Sistem keranjang

Pada halaman keranjang dapat menampung produk-produk yang akan dibeli dalam bentuk tabel dan dapat di produk tersebut bisa dihapus dari tabel keranjang.

Berikut adalah contoh script pada bagian model dengan file bernama Cart\_model yang berisi perintah untuk mengambil ip pengunjung yang kemudian ip tersebut dipakai untuk query pada fungsi getcart() dimana query tersebut berfungsi untuk menampilkan jumlah produk pada keranjang sesuai dengan ip pembeli.

|  |
| --- |
| <?php  class Cart\_model extends CI\_Model{  function ambil\_ip\_pengunjung() {         if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])) {             $ip = $\_SERVER['HTTP\_CLIENT\_IP'];         } elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {             $ip = $\_SERVER['HTTP\_X\_FORWARDED\_FOR'];         } else {             $ip = $\_SERVER['REMOTE\_ADDR'];           return $ip;    public function getcart(){         $ip = $this->ambil\_ip\_pengunjung();     $query = "select sum(jumlah) as jumlah from keranjang where ip = '$ip'";         $result = $this->db->query($query)->row\_array();         $cart = $result['jumlah'];         $result\_cart = 0;         if ($cart == 0){             $result\_cart = 0;         else{             $result\_cart = $cart;           return $result\_cart;  }  }  ?> |

Source code x.x mengambil data jumlah pengunjung

Berikut adalah contoh script pada bagian model dengan file bernama Keranjang\_Model yang berisi perintah untuk menambah produk ke dalam database keranjang sesuai dengan ip pembeli kemudian terdapat query untuk menampilkan data keranjang dan terdapat query untuk menghapus data pada keranjang.

|  |
| --- |
| <?php if (!defined('BASEPATH')) exit('No direct script access allowed');  class Keranjang\_Model extends CI\_Model {     public function get\_kategori\_all()           $query = $this->db->get('produk');         return $query->result\_array();     }     function ambil\_ip\_pengunjung() {             if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])) {                 $ip = $\_SERVER['HTTP\_CLIENT\_IP'];             } elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {                 $ip = $\_SERVER['HTTP\_X\_FORWARDED\_FOR'];             } else {                 $ip = $\_SERVER['REMOTE\_ADDR'];               return $ip;     }     public function addToCart($kode,$jumlah){       if($jumlah == NULL){        $jumlah =1;  $cart =["ip" => $this->ambil\_ip\_pengunjung(),"kode\_produk" => $kode,"jumlah" => $jumlah];        return $this->db->insert('keranjang',$cart);        else{  $cart =["ip" => $this->ambil\_ip\_pengunjung(),"kode\_produk" => $kode,"jumlah" => $jumlah];        return $this->db->insert('keranjang',$cart);        }       public function tampil($kode = 0){         $ip = $this->ambil\_ip\_pengunjung();         $query = "             select                 DISTINCT                 keranjang.id as kode,                 keranjang.kode\_produk as kodeproduk,                 produk.nama\_produk as nama,                 produk.harga\_produk as harga,                 sum(keranjang.jumlah) as jumlah,                 sum(produk.harga\_produk) as total             from             produk             join             keranjang             on             produk.kode\_produk = keranjang.kode\_produk             WHERE keranjang.ip =  '$ip'             group by produk.kode\_produk";           return $this->db->query($query)->result\_array();       public function hapus($kode){         return $this->db->delete('keranjang',['kode\_produk' =>$kode]);       public function hapusemua(){         $ip = $this->ambil\_ip\_pengunjung();  $sql= $this->db->query("delete from keranjang where ip ='$ip'");         return $sql;  }  ?> |

Source code x.x query pada keranjang model

Berikut contoh script pada controller dengan file bernama Marketplace di dalam file tersebut sisipkan code seperti di bawah ini berufungsi untuk mengambil ip pembeli

|  |
| --- |
| function ambil\_ip\_pengunjung() {             if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])) {                 $ip = $\_SERVER['HTTP\_CLIENT\_IP'];             } elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {                 $ip = $\_SERVER['HTTP\_X\_FORWARDED\_FOR'];             } else {                 $ip = $\_SERVER['REMOTE\_ADDR'];               return $ip; |

Source code x.x fungsi mengambil data ip

Kemudian sisipkan code dibawah ini yang berfungsi untuk memasukkan produk ke dalam keranjang.

|  |
| --- |
| public function addToCart(){  $ip = $this->ambil\_ip\_pengunjung();  $kode = $this->input->post('kode\_produk');  $jumlah = $this->input->post('jumlah');  $query = "select stok\_produk from produk where kode\_produk ='$kode'";  $stok = $this->db->query($query)->result\_array();  $this->cart->insert(array(  "id" => $kode,  "jumlah" => $jumlah));  $err = FALSE;  if($ip){  foreach($stok as $s){  if($s['stok\_produk']<=0){  $this->session->set\_flashdata('stok','<div class="alert alert-danger"  role="alert">Maaf Stok Habis</div>');  redirect('marketplace','refresh');  }else{  if($this->Keranjang\_Model->addToCart($kode,$jumlah)) {  $this->session->set\_flashdata('pesan','<div class="alert alert-success"  role="alert"> Berhasil ditambahkan ke dalam cart</div>');}  if(!$err){  $this->session->set\_flashdata('pesan','<div class="alert alert-danger"  role="alert"> Gagal ditambahkan ke dalam cart</div>');}  $tujuan = 'marketplace';  redirect($tujuan);  }}}} |

Source code x.x menambah produk kedalam keranjang

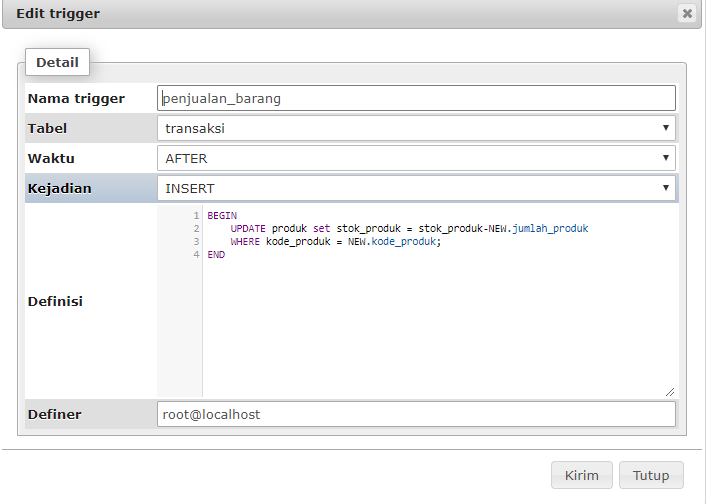
Berikut contoh script pada controller dengan file yang bernama Keranjang yaitu berisi perintah untuk menampilkan data keranjang yang nantinya akan ditampilkan dalam bentuk tabel dan juga terdapat fungsi untuk menghapus data keranjang.

|  |
| --- |
| <?php     Class Keranjang extends CI\_Controller           private $data;         public function \_\_construct()      {          parent::\_\_construct();          $this->load->library('cart');          //$this->load->model('Marketplace\_model');         $this->load->model('Keranjang\_Model');         $this->load->model('cart\_model');         $this->data['data'] = $this->cart\_model->getcart();         $this->data['judul'] = 'KaktusKmi';         public function tampil\_cart()           $data['judul'] = 'Keranjang';  $data['kategori']= $this->Keranjang\_Model->get\_kategori\_all();         $data['data'] = $this->Keranjang\_Model->tampil();           $this->load->view('templates/header',$this->data);         $this->load->view('Keranjang/tampil\_cart',$data);         $this->load->view('templates/footer');       public function hapus($kode){           if($this->Keranjang\_Model->hapus($kode)){  $this->session->set\_flashdata('pesan','<div class="alert alert-success" role="alert">"pesan di hapus"</div>');         else{  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">"pesan gagal dihapus"</div>');           redirect('Keranjang/tampil\_cart');     }    ?> |

Source code x.x controller keranjang

1. Sabtu, 18 Mei 2019

Kegiatan : Data stok berkurang saat pembeli membeli produk

Untuk mengurangi data stok pada saat pembeli membeli produk yaitu dengan menambahkan trigger pada phpmyadmin seperti gambar dibawah ini.

Gambar x.x trigger stok produk

1. Minggu – Selasa, 23 - 25 Mei 2019

Kegiatan : membuat logic pembelian

Pada proses pembelian terdapat 3 status yaitu menunggu ongkir, menunggu pembayaran dan terkonfirmasi. Pada saat status menunggu ongkir admin menginputkan biaya ongkir yang ditanggung pembeli kemudian otomatis status berubah ke menunggu pembayaran kemudian pembeli mengupload bukti pembayarannya, di bagian admin akan ada keterangan bahwa pembeli sudah mengupload bukti pembayaran kemudian admin mengkonfirmasi pembelian maka status akan berubah menjadi terkonfirmasi.

Berikut contoh script pada controller dengan file yang bernama Pembelian yaitu berisi perintah untuk menampilkan halaman pada status yang berbeda-beda.

|  |
| --- |
| public function menunggu\_ongkir($id)  {  $data['ongkir'] =$this->Pembelian\_Model->Model\_ongkir($id);  $data['judul'] = 'Halaman Status Pengiriman';  $data['tampil'] = $this->Pembelian\_Model->tampil($id);  $this->load->view('templates/header', $this->data);  $this->load->view('pembelian/menunggu\_ongkir',$data);  $this->load->view('templates/footer');  }  public function menunggu\_pembayaran($id)  {  $data['judul'] = 'Halaman Status Pengiriman';  if(empty($\_FILES['userfile']['name'])) {  $this->session->set\_userdata('temp\_kode\_pembeli', $id);    $gambare = $this->Pembelian\_Model->upload\_gambar();  // echo $gambare['result']; die;  if($gambare['result'] != 'failed') {  if($this->Pembelian\_Model->save\_gambar($gambare) > 0) {  $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Bukti pembayaran berhasil diupload</div>');  } else {  $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">Bukti pembayaran gagal diupload</div>');  }  }  }  $ongkir = $this->Pembelian\_Model->Model\_ongkir($id);    $data['kode\_pembeli'] = $id;  $data['ongkir'] = $ongkir;    $data['tampil'] = $this->Pembelian\_Model->tampil($id);    $this->load->view('templates/header', $this->data);  $this->load->view('pembelian/menunggu\_pembayaran',$data);  $this->load->view('templates/footer');  }  public function terkonfirmasi($id)  {  $data['judul'] = 'Halaman Status Pengiriman';  $data['ongkir'] =$this->Pembelian\_Model->Model\_ongkir($id);  $data['tampil'] = $this->Pembelian\_Model->tampil($id);  $this->load->view('templates/header', $this->data);  $this->load->view('pembelian/terkonfirmasi',$data);  $this->load->view('templates/footer');  } |

Source code x.x controller pembelian

Berikut contoh script pada controller dengan file yang bernama Konfirmasi yaitu berisi perintah untuk mengatur status pada pembelian contohnya jika status menunggu ongkir maka menuju halaman menunggu ongkir.

|  |
| --- |
| <?php  Class Konfirmasi extends CI\_Controller  {  private $data;  public function \_\_construct(){  parent::\_\_construct();  $this->load->model('cart\_model');  $this->load->model('Konfirmasi\_Model');  $this->load->library('form\_validation');  $this->data['data'] = $this->cart\_model->getcart();  $this->data['judul'] = 'KaktusKmi';    }  public function index()  {  $data['judul'] = 'Konfirmasi Pembayaran';  $this->form\_validation->set\_rules('nomor', 'Nomor', 'required|trim|numeric', [  'required' => 'Nomor Harus Di isi',  'numeric' => 'Nomor harus angka'  ]);  $this->form\_validation->set\_rules('kode', 'Kode', 'required|min\_length[5]', [  'required' => 'Kode Harus Di isi',  'min\_length' => 'Kode tidak valid'  ]);  if($this->form\_validation->run() == FALSE) {  $this->load->view('templates/header', $this->data);  $this->load->view('konfirmasi/index',$data);  $this->load->view('templates/footer');  } else {  $dt = [  'nomor' => $this->input->post('nomor'),  'kode' => $this->input->post('kode')  ];  $status = $this->Konfirmasi\_Model->getStatus($dt);  if($status) {  $statusnya = $status->status;  if($statusnya == "menunggu ongkir") {  redirect('Pembelian/menunggu\_ongkir/'.$status->id\_pembeli);  } else if($statusnya == "menunggu pembayaran") {  redirect('Pembelian/menunggu\_pembayaran/'.$status->id\_pembeli);  } else {  redirect('Pembelian/terkonfirmasi/'.$status->id\_pembeli);  }  } else {  $this->session->set\_flashdata('pesan', "<div class='alert alert-danger' role='alert'>Kode Pesanan tidak ada.</div>");  redirect('Konfirmasi/index');  }  }  }  }  ?> |

Source code x.x controller konfirmasi

Berikut contoh script pada controller dengan file yang bernama Verifikasi pada bagian admin yaitu berisi perintah untuk mengubah status pembelian dan menambahkan biaya ongkir.

|  |
| --- |
| public function verification\_admin()  {  $data['judul'] = 'Halaman Verifikasi';  $data['pembeli'] = $this->Verifikasi\_Model->dapatkanSemuaDataPembeli();  $this->load->view('templates/header\_admin', $data);  $this->load->view('adminpage/verifikasi/verification\_admin');  $this->load->view('templates/footer');  }  public function ongkir\_verification($kode)  {  $data['judul'] = 'Halaman Detail Verifikasi';  $this->form\_validation->set\_rules('biaya', 'Biaya', 'required|numeric|trim', [  'required' => 'Kudu diisi coy',  'numeric' => 'Biaya nya harus angka bosque'  ]);  if($this->form\_validation->run() == FALSE) {  $data['tampil'] = $this->Verifikasi\_Model->tampil($kode);  $data['pembeli'] = $this->Verifikasi\_Model->dapatkanSatuDataPembeli($kode);    $this->load->view('templates/header\_admin', $data);  $this->load->view('adminpage/verifikasi/ongkir\_verification',$data);  $this->load->view('templates/footer');  } else {  $dt = [  'kode' => $kode,  'biaya' => $this->input->post('biaya')  ];  if($this->Verifikasi\_Model->ubahStatusOngkir($dt)) {  $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Status berhasil diubah</div>');  } else {  $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">Status gagal diubah</div>');  }  redirect('adminpage/Verifikasi/verification\_admin');  }  }  public function konfirmasi\_verification()  {  $data['judul'] = 'Halaman Detail Verifikasi';  $this->load->view('templates/header\_admin', $data);  $this->load->view('adminpage/verifikasi/konfirmasi\_verification');  $this->load->view('templates/footer');  }  public function ubahStatusTerkonfirmasi($kode) {  $hasil = $this->Verifikasi\_Model->ubahStatus($kode);  if( $hasil > 0) {  $this->session->set\_flashdata('pesan', '<div class="alert alert-success" role="alert">Status berhasil diubah</div>');  } else {  $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">Status gagal diubah karena belum memberikan bukti pembayaran</div>');  } redirect('adminpage/Verifikasi/verification\_admin');  } |

Source code x.x controller verifikasi

1. Dokumentasi



Gambar x.x kegiatan pengerjaan mtp