

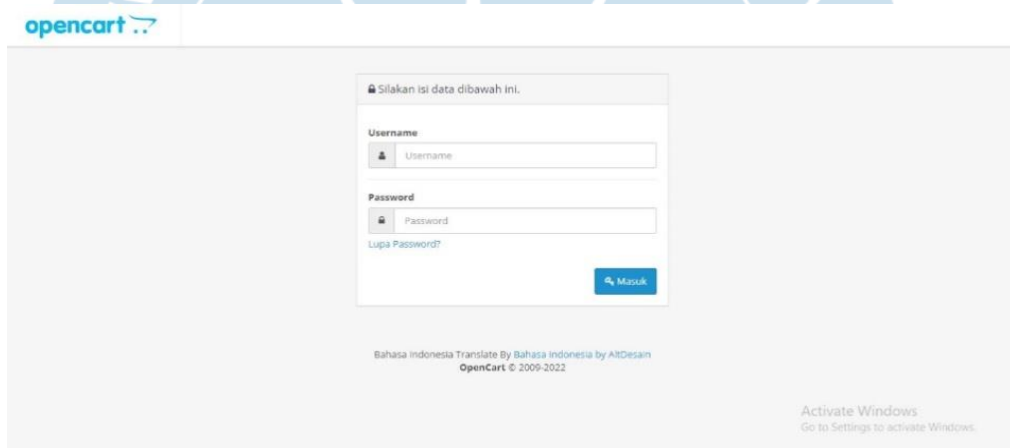
## BAB V. IMPLEMENTASI DAN PENGUJIAN SISTEM

### 5.1. Implementasi Sistem

#### 5.1.1 Implementasi *Login Admin*

Implementasi pada gambar 5.1 digunakan oleh admin untuk masuk kedalam *website* ICKomp dalam mengelola data toko. Admin memasukkan *username* dan *password* kemudian akan menuju halaman beranda bagi admin.

Pada gambar 5.2 merupakan potongan *code* yang menangani proses *login* pada sistem. Fungsi *login* pada *LoginController* digunakan untuk melakukan pengecekan kepada basis data. Jika data ditemukan maka pesan respon yang dikembalikan sistem adalah ok dan jika tidak ditemukan, maka pesan respon adalah *failed*.



Gambar 5.1 Implementasi *Login Admin*

```
class ControllerAccountLogin extends Controller {
    private $error = array();

    public function index() {
        $this->load->model('account/customer');

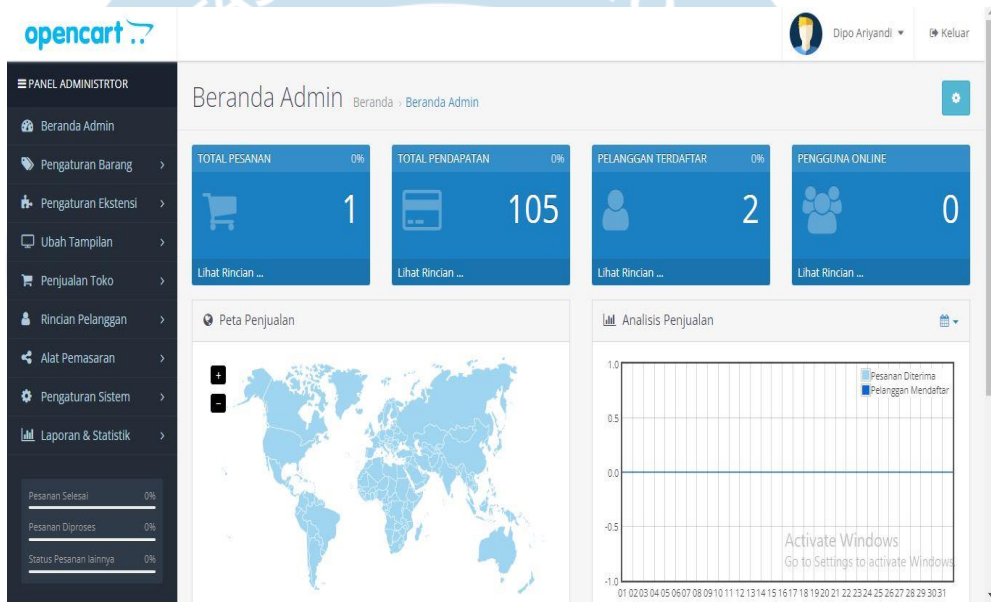
        // Login override for admin users
        if (!empty($this->request->get['token'])) {
            $this->customer->logout();
            $this->cart->clear();
        }
    }
}
```

Gambar 5.2 *Login Admin Controller*

### 5.1.2 Implementasi Beranda Admin

Pada gambar 5.3 menampilkan halaman beranda admin dimana menampilkan rincian data toko dan proses pesanan yang ada di ICKomp. Admin dapat mengelola semua data toko maupun dapat melihat proses pesanan yang sedang berjalan yang dilakukan oleh pembeli.

Pada gambar 5.4 merupakan potongan *code* yang menangani proses dihalaman beranda admin. Fungsi *index()* pada *dashbordcontroller* digunakan untuk menampilkan halaman beranda admin dimana admin dapat mengelola *website* toko.



Gambar 5.3 Implementasi Halaman Beranda Admin

```
class ControllerCommonDashboard extends Controller {
    public function index() {
        $this->load->language('common/dashboard');

        $this->document->setTitle($this->language->get('heading_title'));

        $data['user_token'] = $this->session->data['user_token'];

        $data['breadcrumbs'] = array();

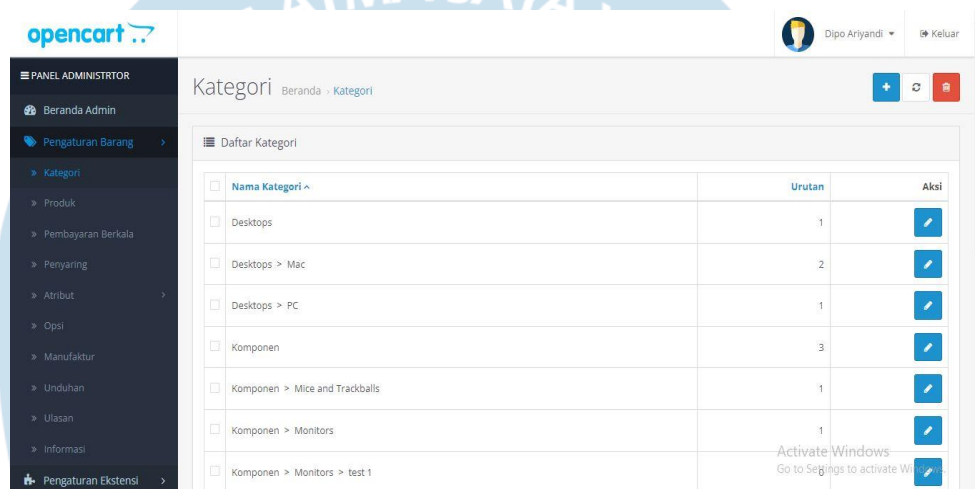
        $data['breadcrumbs'][] = array(
            'text' => $this->language->get('text_home'),
            'href' => $this->url->link('common/dashboard', 'user_token=' . $this->session->data['user_token'], true)
        );
    }
}
```

Gambar 5.4 Halaman Beranda Admin Controller

### 5.1.3 Implementasi Halaman Kategori Produk

Pada gambar 5.5 menampilkan halaman pengaturan kategori produk, dimana admin dapat mengolah kategori produk yang ada di ICKomp. Admin dapat menambahkan, mengedit maupun menghapus kategori produk.

Pada gambar 5.6 merupakan potongan *code* yang menangani proses pengelolaan kategori produk. Fungsi *getcategory()* pada *ModelCatalogCategory* menampilkan halaman kategori produk yang dapat dilihat oleh admin dan admin dapat melakukan proses tambah, edit dan hapus kategori produk.



Gambar 5.5 Implementasi Halaman Beranda Admin

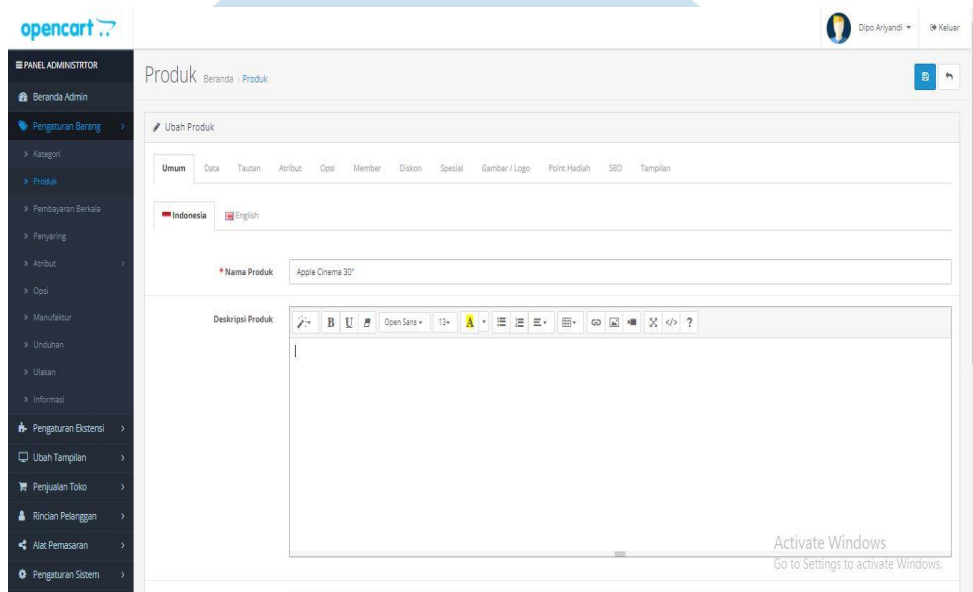
```
<?php
class ModelCatalogCategory extends Model {
    public function getCategory($category_id) {
        $query = $this->db->query("SELECT DISTINCT * FROM " . DB_PREFIX . "category c LEFT JOIN " . DB_
        return $query->row;
    }
}
```

Gambar 5.6 Halaman Beranda Admin Controller

### 5.1.4 Implementasi Halaman Ubah Kategori

Pada gambar 5.7 menampilkan halaman ubah kategori produk dimana admin dapat mengubah detail kategori produk yang akan ditampilkan pada halaman *website* ICKomp. Admin dapat memasukkan nama dan deskripsi produk sebelum diunggah.

Pada gambar 5.8 merupakan potongan *code* yang menangani proses mengubah kategori produk yang dilakukan sistem ketika admin mengeklik menu ubah.



Gambar 5.7 Implementasi Halaman Ubah Kategori

```
public function edit() {
    $this->load->language('catalog/category');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('catalog/category');

    if (($this->request->server['REQUEST_METHOD'] == 'POST') && $this->validateForm()) {
        $this->model_catalog_category->editCategory($this->request->get['category_id'], $this->request->post);

        $this->session->data['success'] = $this->language->get('text_success');

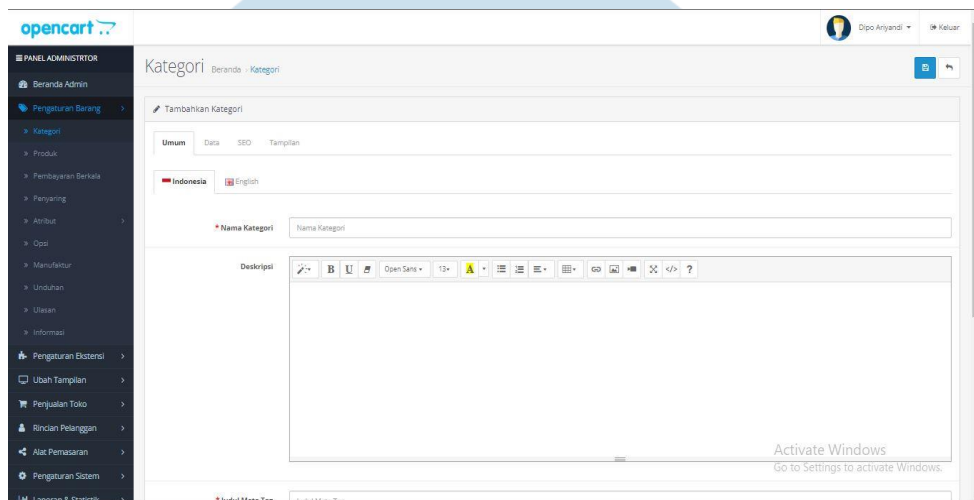
        $url = '';
    }
}
```

Gambar 5.8 Halaman Ubah Kategori *controller*

### 5.1.5 Implementasi Halaman Tambah Kategori

Pada gambar 5.9 menampilkan halaman tambah kategori produk dimana admin dapat menambahkan kategori baru dengan memasukkan nama dan deskripsi kategori untuk ditampilkan pada halaman *website* ICKomp.

Pada gambar 5.10 merupakan potongan *code* yang menangani proses tambah produk. Jika kategori yang ditambahkan valid maka fungsi *addcategory* akan berjalan dan menambahkan kategori baru pada *website*.



Gambar 5.9 Implementasi Halaman Tambah Kategori

```
public function add() {
    $this->load->language('catalog/category');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('catalog/category');

    if (($this->request->server['REQUEST_METHOD'] == 'POST') && $this->validateForm()) {
        $this->model_catalog_category->addCategory($this->request->post);

        $this->session->data['success'] = $this->language->get('text_success');

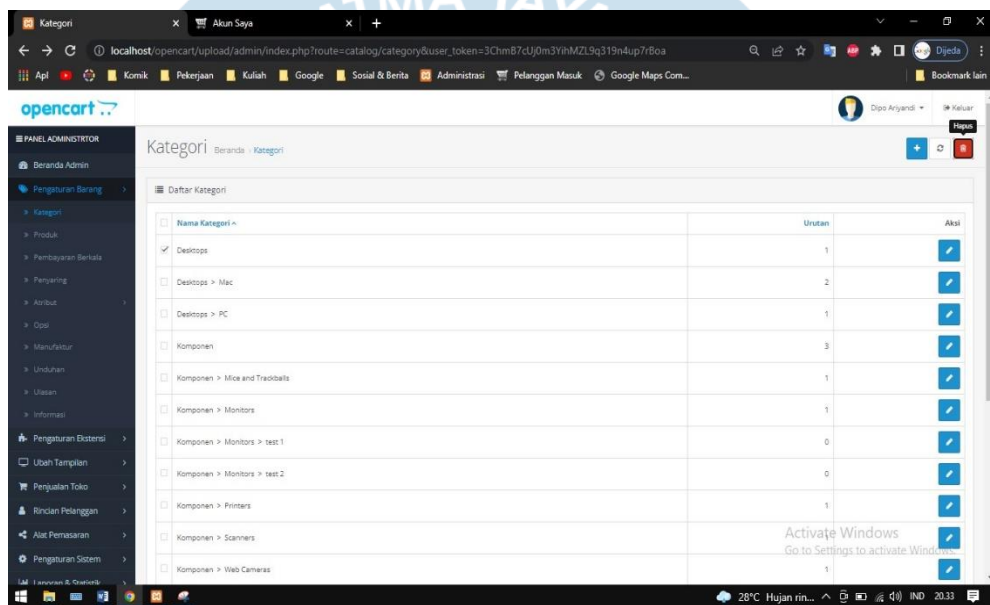
        $url = '';
    }
}
```

Gambar 5.10 Halaman Tambah Kategori *controller*

### 5.1.6 Implementasi Halaman Hapus Kategori

Pada gambar 5.11 menampilkan halaman hapus kategori produk dimana admin dapat menghapus deskripsi maupun data kategori dari produk-produk yang akan dijual di halaman *website* ICKomp.

Pada gambar 5.12 merupakan potongan *code* yang menampilkan proses hapus kategori. Ketika admin memilih kategori yang ingin dihapus kemudian mengklik hapus maka fungsi *deletecategory* pada *model\_catalog\_category* akan berjalan.



Gambar 5.11 Implementasi Halaman Hapus Kategori

```
public function delete() {
    $this->load->language('catalog/category');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('catalog/category');

    if (isset($this->request->post['selected']) && $this->validateDelete()) {
        foreach ($this->request->post['selected'] as $category_id) {
            $this->model_catalog_category->deleteCategory($category_id);
        }
    }

    $this->session->data['success'] = $this->language->get('text_success');

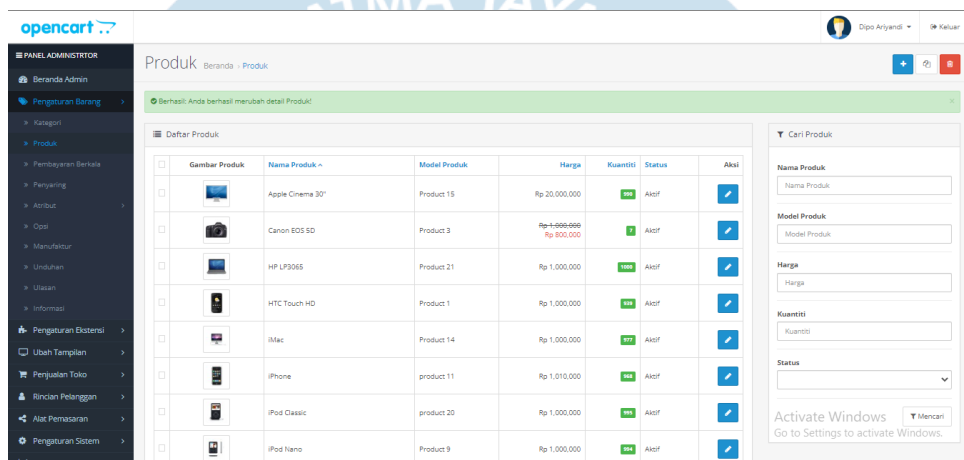
    $url = '';
}
```

Gambar 5.12 Halaman Hapus Kategori Controller

### 5.1.7 Implementasi Halaman Pengaturan Produk

Pada gambar 5.13 menampilkan halaman pengaturan produk dimana admin dapat melihat detail produk yang ada di ICKomp. Admin dapat mengatur detail produk baik menambahkan, mengedit maupun menghapus produk.

Pada gambar 5.14 merupakan potongan *code* yang menampilkan proses pengaturan produk dimana fungsi *updateviewed* pada *modelcatalogproduk* akan menampilkan halaman pengaturan produk pada admin.



Gambar 5.13 Implementasi Halaman Pengaturan Produk

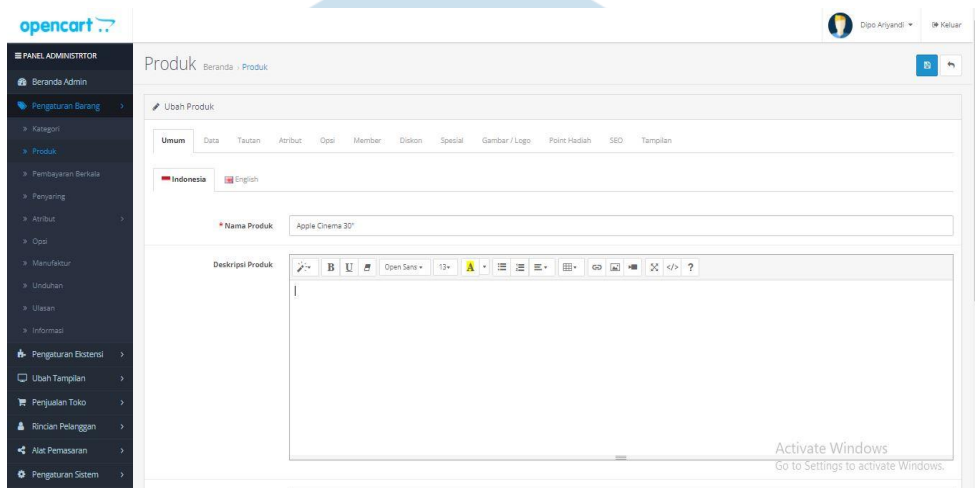
```
class ModelCatalogProduct extends Model {  
    public function updateViewed($product_id) {  
        $this->db->query("UPDATE " . DB_PREFIX . "product SET viewed = (viewed + 1) WHERE product_id = " . $product_id . ";");  
    }  
}
```

Gambar 5.14 Halaman Pengaturan Produk *controller*

### 5.1.8 Implementasi Halaman Ubah Produk

Pada gambar 5.15 menampilkan halaman ubah produk dimana dapat mengubah data produk yang telah ada untuk diperbarui, admin dapat mengubah nama maupun deskripsi produk yang sudah ada.

Pada gambar 5.16 merupakan potongan *code* yang menampilkan proses ubah produk, dimana fungsi *edit()* pada *model\_catalog\_product* akan berjalan



Gambar 5.15 Implementasi Halaman Ubah Produk

```
public function edit() {
    $this->load->language('catalog/product');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('catalog/product');

    if (($this->request->server['REQUEST_METHOD'] == 'POST') && $this->validateForm()) {
        $this->model_catalog_product->editProduct($this->request->get['product_id'], $this->request->post);

        $this->session->data['success'] = $this->language->get('text_success');

        $url = '';
    }
}
```

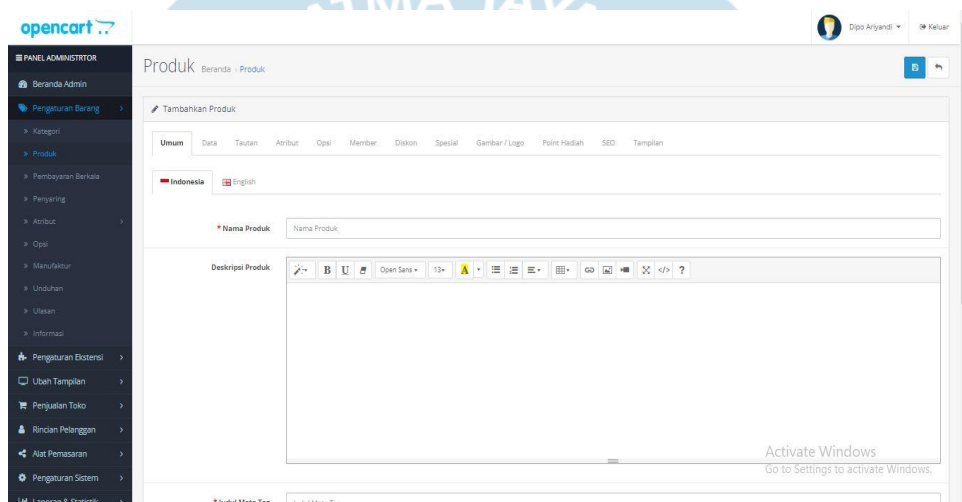
Gambar 5.16 Halaman Ubah Produk Controller



### 5.1.9 Implementasi Halaman Tambah Produk

Pada gambar 5.17 menampilkan halaman tambah produk dimana admin dapat menambah produk baru yang belum ada pada toko, admin dapat memasukkan nama dan deskripsi produk baru untuk ditampilkan pada halaman produk.

Pada gambar 5.18 merupakan potongan *code* yang menampilkan proses penambahan produk. Jika kategori yang ditambahkan valid maka fungsi *addproduct* akan berjalan dan menambahkan kategori baru pada *website*.



Gambar 5.17 Implementasi Halaman Tambah Produk

```
public function add() {
    $this->load->language('catalog/product');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('catalog/product');

    if (($this->request->server['REQUEST_METHOD'] == 'POST') && $this->validateForm()) {
        $this->model_catalog_product->addProduct($this->request->post);

        $this->session->data['success'] = $this->language->get('text_success');

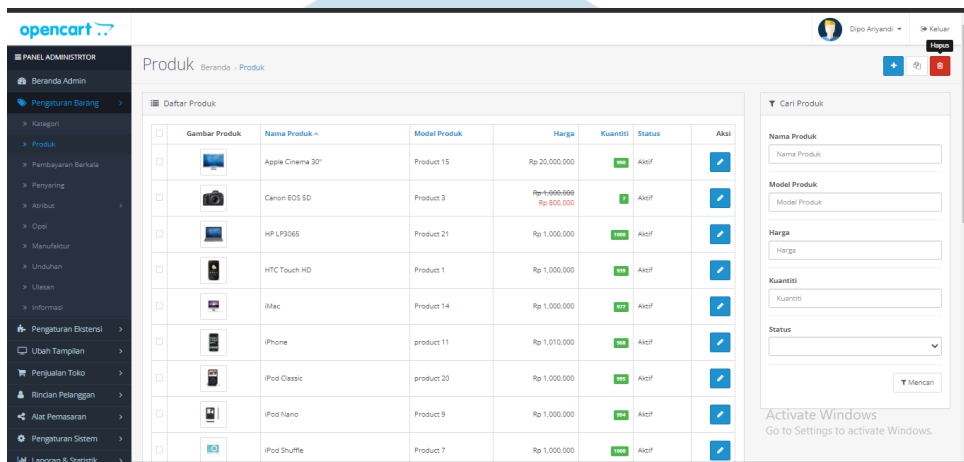
        $url = '';
    }
}
```

Gambar 5.18 Halaman Tambah Produk Controller

### 5.1.10 Implementasi Halaman Hapus Produk

Pada gambar 5.19 menampilkan halaman hapus produk dimana admin dapat menghapus produk yang telah ada di halaman produk.

Pada gambar 5.20 merupakan potongan *code* yang memproses hapus produk dimana fungsi *delete()* pada *catalogproduct* akan berjalan ketika admin memilih produk yang ingin dihapus dan mengklik tombol hapus.



Gambar 5.19 Implementasi Halaman Hapus Produk

```
public function delete() {
    $this->load->language('catalog/product');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('catalog/product');

    if (isset($this->request->post['selected']) && $this->validateDelete()) {
        foreach ($this->request->post['selected'] as $product_id) {
            $this->model_catalog_product->deleteProduct($product_id);
        }

        $this->session->data['success'] = $this->language->get('text_success');

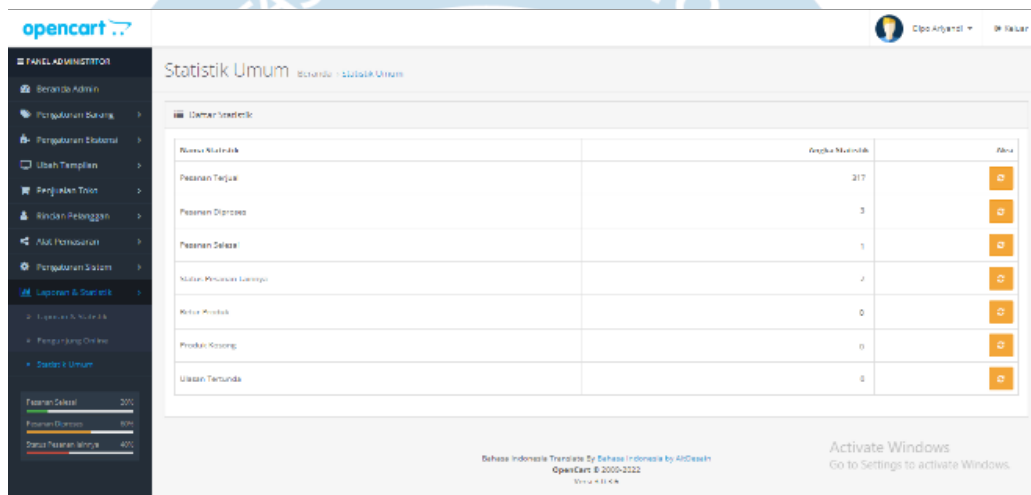
        $url = '';
    }
}
```

Gambar 5.20 Halaman Hapus Produk Controller

### 5.1.11 Implementasi Halaman Statistik Umum

Pada gambar 5.21 menampilkan halaman statistik umum dimana admin dapat melihat semua rincian pesanan yang masuk di ICKomp. Admin dapat melihat pesanan yang telah terselesaikan, maupun pesanan yang masih dalam proses transaksi.

Pada gambar 5.22 merupakan potongan *code* yang berjalan ketika admin mengklik menu laporan dan statistik dimana fungsi *getstatistics* pada *modelreportstatistics* digunakan untuk menampilkan laporan dan statistik toko.



Rincian Statistik	Angka Statistik	Aksi
Pesanan Terjual	217	[Aksi]
Pesanan Dibatasi	0	[Aksi]
Pesanan Dibatasi	1	[Aksi]
Nota Pinjaman Lunas	7	[Aksi]
Kerita Pinjaman	0	[Aksi]
Produk Kosong	0	[Aksi]
Ulasan Tertunda	0	[Aksi]

Gambar 5.21 Implementasi Halaman Statistik Umum

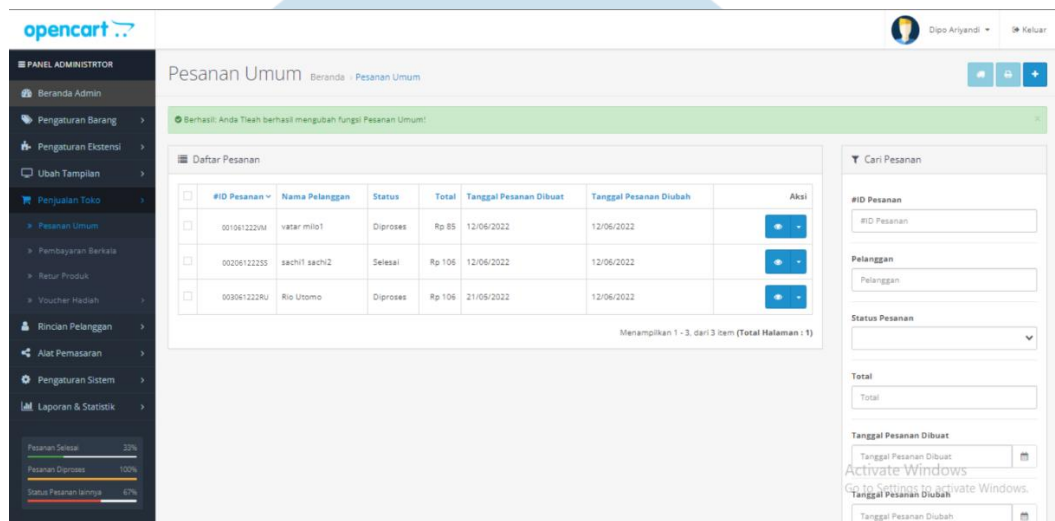
```
class ModelReportStatistics extends Model {  
    public function getStatistics() {  
        $query = $this->db->query("SELECT * FROM " . DB_PREFIX . "statistics");  
        return $query->rows;  
    }  
}
```

Gambar 5.22 Halaman Statistik Umum Controller

### 5.1.12 Implementasi Halaman Pesanan Umum

Pada gambar 5.23 menampilkan halaman pesanan umum bagi admin dimana admin dapat melihat semua data pesanan dari pembeli yang masuk pada ICkomp. Admin dapat melakukan pengecekan status pesanan pembeli sebelum memproses pesanan dari pembeli.

Pada gambar 5.24 merupakan potongan *code* yang berjalan ketika admin ingin melihat data pesanan dari pembeli.



Gambar 5.23 Implementasi Halaman Pesanan Umum

```
array_multisort($sort_order, SORT_ASC, $data['reports']);

if (isset($this->request->get['code'])) {
    $data['report'] = $this->load->controller('extension/report/' . $this->request->get['code'] . '/report');
} elseif (isset($data['reports'][0])) {
    $data['report'] = $this->load->controller('extension/report/' . $data['reports'][0]['code'] . '/report');
} else {
    $data['report'] = '';
}

$data['header'] = $this->load->controller('common/header');
$data['column_left'] = $this->load->controller('common/column_left');
$data['footer'] = $this->load->controller('common/footer');

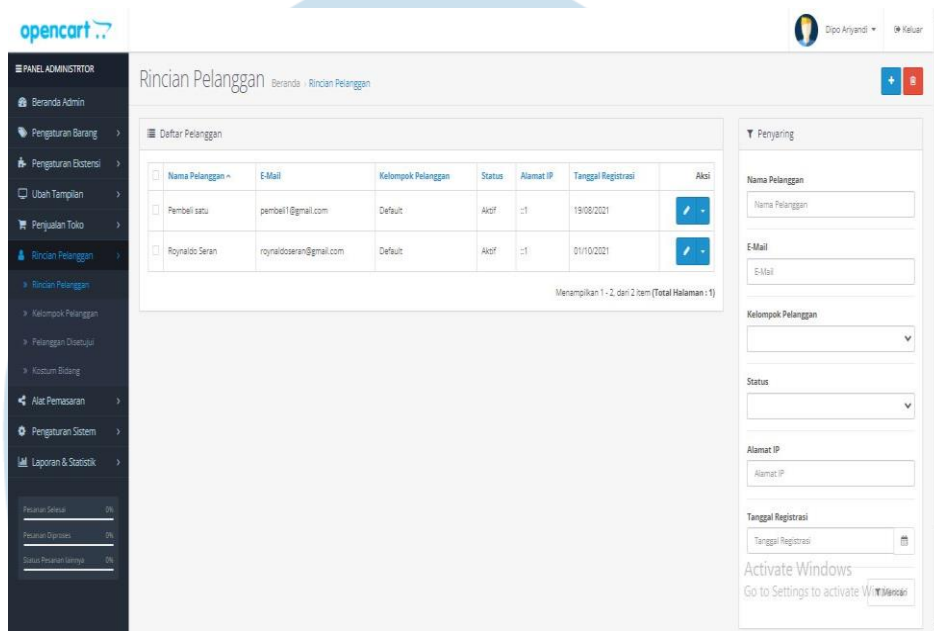
$this->response->setOutput($this->load->view('report/report', $data));
```

Gambar 5.24 Halaman Pesanan Umum Controller

### 5.1.13 Implementasi Halaman Rincian Pelanggan

Pada gambar 5.25 menampilkan halaman rincian pelanggan bagi admin dimana admin dapat melihat data pelanggan yang telah membuat akun pada *website* ICKomp. Admin dapat mengubah, mengedit dan menghapus data pembeli.

Pada gambar 5.26 merupakan potongan *code* yang berjalan pada sistem saat admin ingin melihat rincian pelanggan.



Gambar 5.25 Implementasi Halaman Rincian Pelanggan

```
public function getCustomFields($customer_group_id = 0) {
    $custom_field_data = array();

    if (!$customer_group_id) {
        $custom_field_query = $this->db->query("SELECT * FROM `". DB_PREFIX . "custom_field` cf LEFT JOIN `".
    } else {
        $custom_field_query = $this->db->query("SELECT * FROM `". DB_PREFIX . "custom_field_customer_group` cfc");
    }

    foreach ($custom_field_query->rows as $custom_field) {
        $custom_field_value_data = array();

        if ($custom_field['type'] == 'select' || $custom_field['type'] == 'radio' || $custom_field['type'] == 'checkbox') {
            $custom_field_value_query = $this->db->query("SELECT * FROM `". DB_PREFIX . "custom_field_value` cfv");

            foreach ($custom_field_value_query->rows as $custom_field_value) {
                $custom_field_value_data[] = array(
                    'custom_field_value_id' => $custom_field_value['custom_field_value_id'],
                    'name' => $custom_field_value['name']
                );
            }
        }
    }
}
```

Gambar 5.26 Halaman Rincian Pelanggan Controller

### 5.1.14 Implementasi Halaman Daftar Akun

Pada gambar 5.27 menampilkan halaman daftar akun bagi pembeli dimana pembeli dapat mendaftar akun sebagai pelanggan toko ICKomp. Pembeli yang telah mendaftar dapat *login* ke *website* ICKomp dengan email dan *password* yang telah dibuat atau didaftarkan.

Pada gambar 5.28 merupakan potongan *code* yang menangani proses daftar akun pada sistem. Fungsi *index()* pada *controlleraccountregister* digunakan untuk menyimpan data pelanggan kedalam tabel pelanggan.

Gambar 5.27 Implementasi Halaman Daftar Akun

```
class ControllerAccountRegister extends Controller {
    private $error = array();

    public function index() {
        if ($this->customer->isLoggedIn()) {
            $this->response->redirect($this->url->link('account/account', '', true));
        }

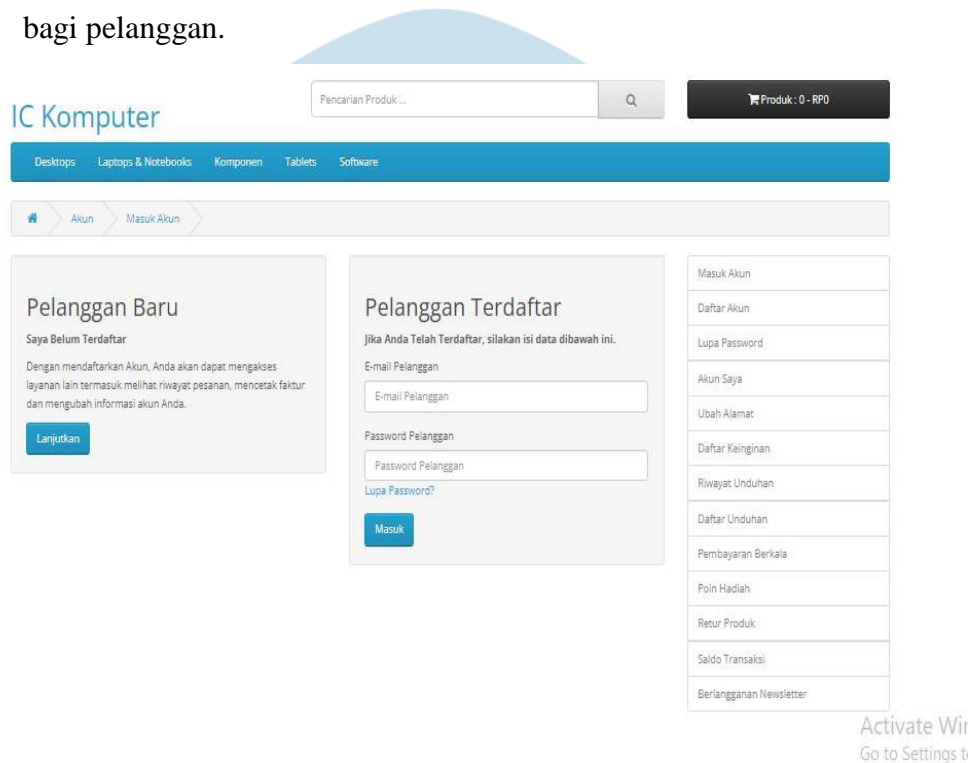
        $this->load->language('account/register');
    }
}
```

Gambar 5.28 Halaman Daftar Akun Controller

### 5.1.15 Implementasi Halaman *Login* Pelanggan

Pada gambar 5.29 menampilkan halaman *login* pelanggan bagi pembeli yang sebelumnya sudah mendaftar akun dan pelanggan melakukan *login* terlebih dahulu sebelum melakukan transaksi di *website* ICKomp. Pelanggan dapat *login* menggunakan akun yang telah didaftarkan.

Pada gambar 5.30 merupakan potongan *code* yang menangani proses *login* bagi pelanggan.



Gambar 5.29 Implementasi Halaman *Login* Pelanggan

```
class ControllerCheckoutLogin extends Controller {
    public function index() {
        $this->load->language('checkout/checkout');

        $data['checkout_guest'] = ($this->config->get('config_checkout_guest') && !$this->config->get('config_customer'));

        if (isset($this->session->data['account'])) {
            $data['account'] = $this->session->data['account'];
        } else {
            $data['account'] = 'register';
        }

        $data['forgotten'] = $this->url->link('account/forgotten', '', true);

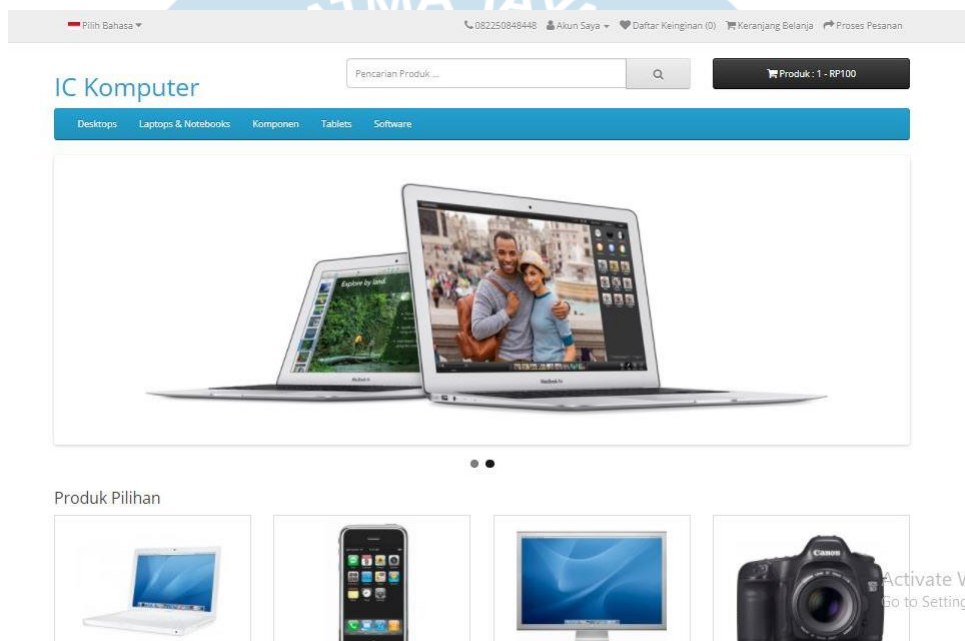
        $this->response->setOutput($this->load->view('checkout/login', $data));
    }
}
```

Gambar 5.30 Halaman *Login* Pelanggan *Controller*

### 5.1.16 Implementasi Halaman Beranda ICKomp

Pada gambar 5.31 menampilkan halaman beranda ICKomp bagi pelanggan, dimana pelanggan melihat tampilan utama dari *website* ICKomp berupa produk-produk yang tersedia. Pelanggan dapat memilih produk berdasarkan kategori yang telah tersedia.

Pada gambar 5.32 merupakan potongan *code* yang menangani proses menampilkan produk toko pada halaman *website* ICKomp bagi pelanggan. Fungsi *index()* pada *controllercommonhome* digunakan untuk menampilkan produk toko ICKomp pada *website*.



Gambar 5.31 Implementasi Halaman Beranda ICKomp

```
class ControllerCommonHome extends Controller {
    public function index() {
        $this->document->setTitle($this->config->get('config_meta_title'));
        $this->document->setDescription($this->config->get('config_meta_description'));
        $this->document->setKeywords($this->config->get('config_meta_keyword'));

        if (isset($this->request->get['route'])) {
            $this->document->addLink($this->config->get('config_url'), 'canonical');
        }
    }
}
```

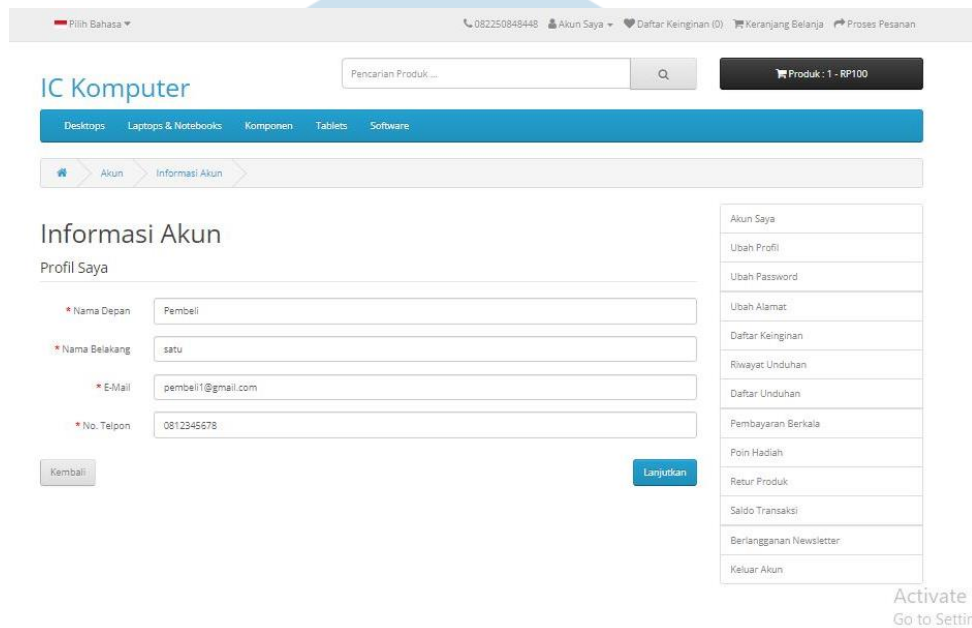
Gambar 5.32 Halaman Beranda ICKomp Controller



### 5.1.17 Implementasi Halaman Informasi Akun

Pada gambar 5.33 menampilkan halaman informasi akun bagi pelanggan dimana pelanggan dapat melihat informasi akun yang telah didaftarkan sebelumnya. Pelanggan dapat mengubah informasi akun apabila terdapat pembaruan data sebelum melakukan transaksi.

Pada Gambar 5.34 merupakan potongan *code* yang menangani proses menampilkan informasi akun pelanggan.



Gambar 5.33 Implementasi Halaman Informasi Akun

```
class ControllerAccountAccount extends Controller {
    public function index() {
        if (!$this->customer->isLoggedIn()) {
            $this->session->data['redirect'] = $this->url->link('account/account', '', true);

            $this->response->redirect($this->url->link('account/login', '', true));
        }

        $this->load->language('account/account');
    }
}
```

Gambar 5.34 Halaman Informasi Akun *Controller*

### 5.1.18 Implementasi Halaman Pengaturan Alamat

Pada gambar 5.35 menampilkan halaman pengaturan alamat bagi pelanggan, dimana pelanggan akan diminta mengatur atau memasukkan alamat yang sesuai sebagai alamat pengiriman apabila melakukan pemesanan produk.

Pada gambar 5.36 merupakan potongan *code* yang menangani proses pengaturan alamat pengguna.

The screenshot shows the 'Tambahkan Alamat' (Add Address) page in the 'IC Komputer' application. The page has a search bar at the top with the text 'Pencarian Produk ...'. Below the search bar is a navigation menu with 'Desktops', 'Laptops & Notebooks', 'Komponen', 'Tablets', and 'Software'. The main navigation bar includes 'Akun', 'Ubah Alamat', and 'Tambahkan Alamat'. The form is titled 'Tambahkan Alamat' and contains the following fields: 'Nama Depan' (Name Front), 'Nama Belakang' (Name Back), 'Nama Perusahaan' (Company Name), 'Alamat Baris 1' (Address Line 1), 'Alamat Baris 2' (Address Line 2), 'Kota' (City), 'Kodepos' (Postal Code), 'Negara' (Country), and 'Wilayah / Provinsi' (Region / Province). There are also radio buttons for 'Jadikan Sebagai Default?' (Make as Default?). A sidebar on the right contains a list of user profile options: 'Akun Saya', 'Ubah Profil', 'Ubah Password', 'Ubah Alamat', 'Daftar Keinginan', 'Riwayat Unduhan', 'Daftar Unduhan', 'Pembayaran Berkala', 'Poin Hadiah', 'Retur Produk', 'Saldo Transaksi', 'Berlangganan Newsletter', and 'Keluar Akun'. At the bottom right, there is a 'Lanjutkan' (Continue) button. The page also has a 'Kembali' (Back) button and an 'Activate Window' notification.

Gambar 5.35 Implementasi Halaman Pengaturan Alamat

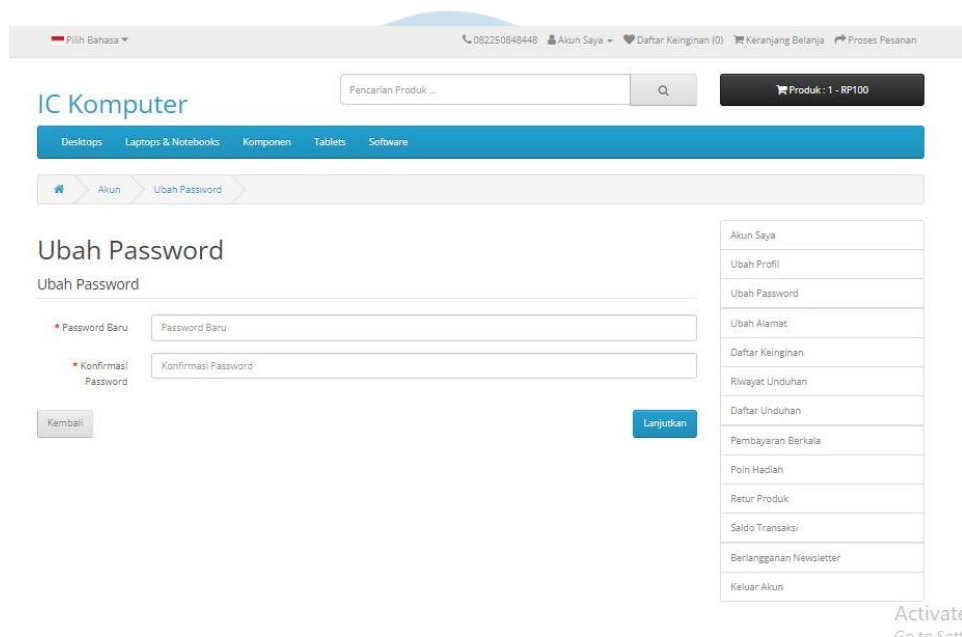
```
public function index() {  
    if (!$this->customer->isLoggedIn()) {  
        $this->session->data['redirect'] = $this->url->link('account/address', '', true);  
        $this->response->redirect($this->url->link('account/login', '', true));  
    }  
  
    $this->load->language('account/address');  
  
    $this->document->setTitle($this->language->get('heading_title'));  
  
    $this->load->model('account/address');  
  
    $this->getList();  
}
```

Gambar 5.36 Halaman Pengaturan Alamat Controller

### 5.1.19 Implementasi Halaman Ubah *Password*

Pada gambar 5.37 menampilkan halaman ubah *password* bagi pelanggan, dimana pelanggan dapat mengubah *password* yang lama dengan yang baru.

Pada gambar 5.38 merupakan potongan *code* yang menangani proses ubah *password* pelanggan.



Gambar 5.37 Implementasi Halaman Ubah *Password*

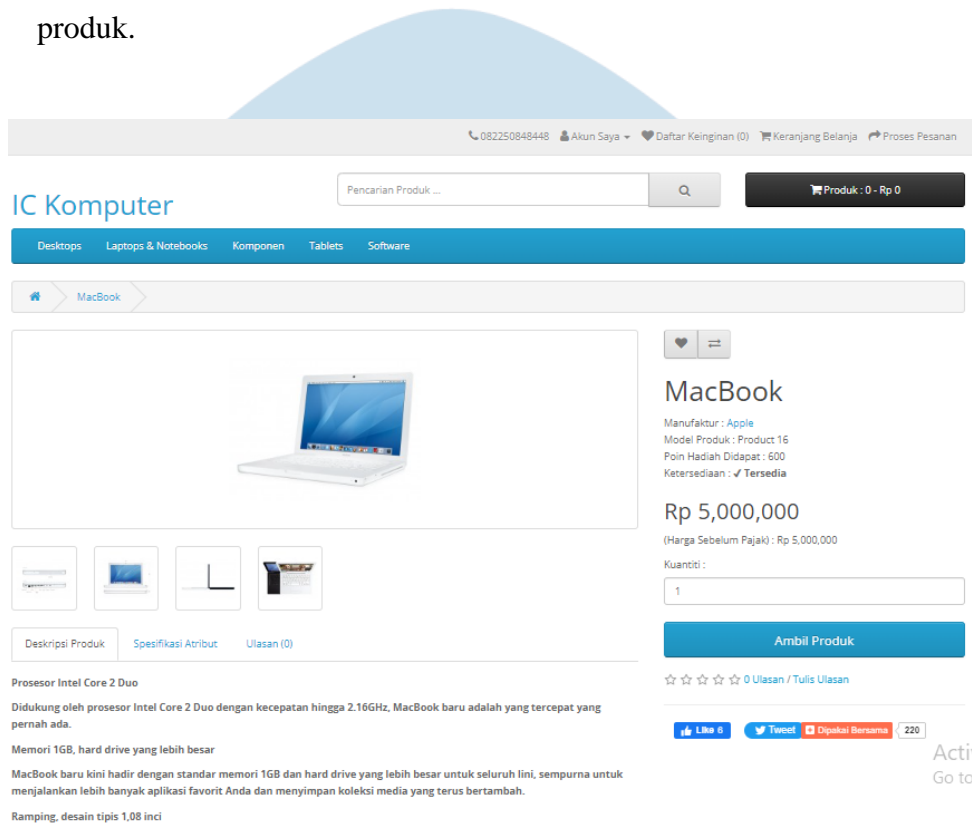
```
$customer_info = $this->model_account_customer->getCustomerByCode($code);  
  
if ($customer_info) {  
    $this->load->language('account/reset');  
  
    $this->document->setTitle($this->language->get('heading_title'));  
  
    if (($this->request->server['REQUEST_METHOD'] == 'POST') && $this->validate()) {  
        $this->model_account_customer->editPassword($customer_info['email'], $this->request->post['password']);  
  
        $this->session->data['success'] = $this->language->get('text_success');  
  
        $this->response->redirect($this->url->link('account/login', '', true));  
    }  
}
```

Gambar 5.38 Halaman Ubah *Password Controller*

### 5.1.20 Implementasi Halaman Pilih Produk

Pada gambar 5.39 menampilkan halaman pilih produk bagi pelanggan, dimana pelanggan dapat melihat produk yang diinginkan sebelum melakukan transaksi. Pelanggan dapat melihat gambar dan deskripsi produk serta memasukkan ke dalam keranjang belanja.

Pada Gambar 5.40 merupakan potongan *code* yang menangani proses pilih produk.



Gambar 5.39 Implementasi Halaman Pilih Produk

```
public function index() {
    if (!$this->customer->isLoggedIn()) {
        $this->session->data['redirect'] = $this->url->link('account/order', '', true);
        $this->response->redirect($this->url->link('account/login', '', true));
    }

    $this->load->language('account/order');

    $this->document->setTitle($this->language->get('heading_title'));

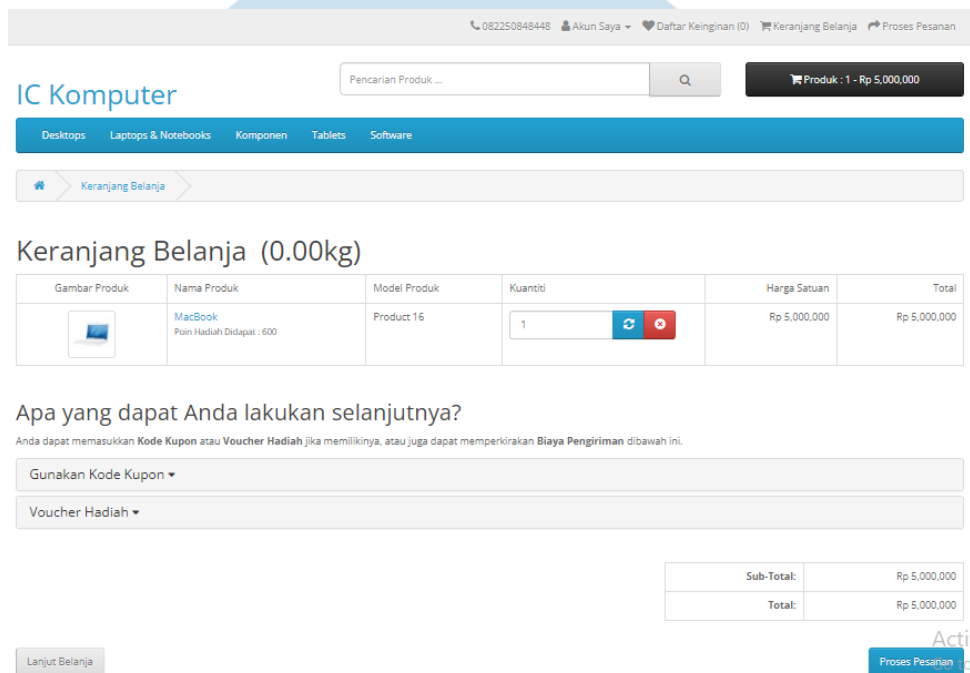
    $url = '';
}
```

Gambar 5.40 Halaman Pilih Produk Controller

### 5.1.21 Implementasi Halaman Keranjang Belanja

Pada gambar 5.41 menampilkan halaman keranjang belanja bagi pelanggan dimana produk yang telah dipilih pelanggan dan dimasukkan ke keranjang akan tertampil di halaman ini. Pelanggan dapat memilih produk apa saja yang ingin diproses pesannya.

Pada Gambar 5.42 merupakan potongan *code* yang menangani proses pada keranjang belanja pelanggan. Fungsi `info()` menampilkan produk yang telah dipilih pelanggan dan masuk ke keranjang.



Gambar 5.41 Implementasi Halaman Keranjang

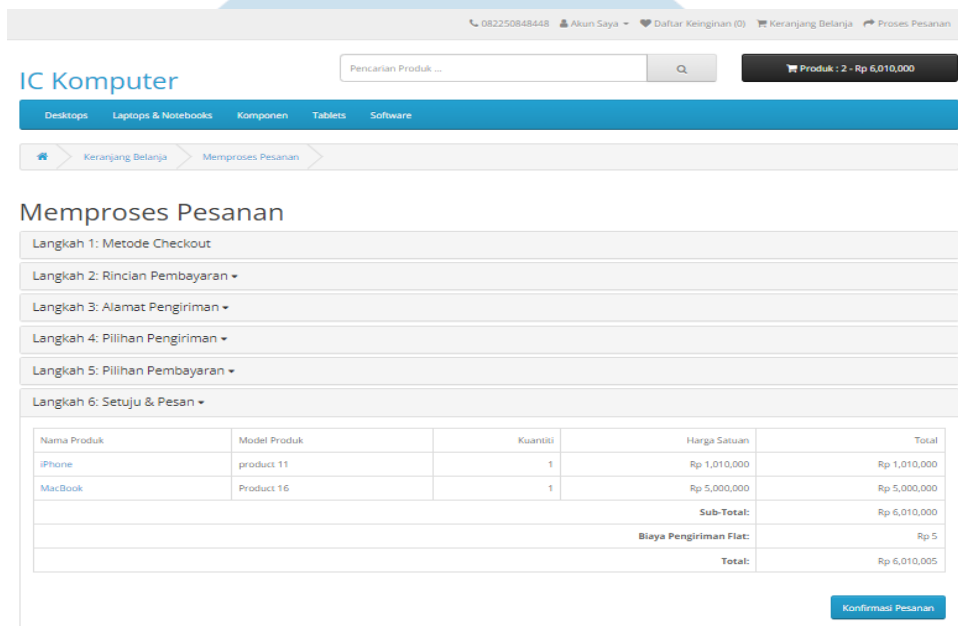


Gambar 5.42 Halaman Keranjang Controller

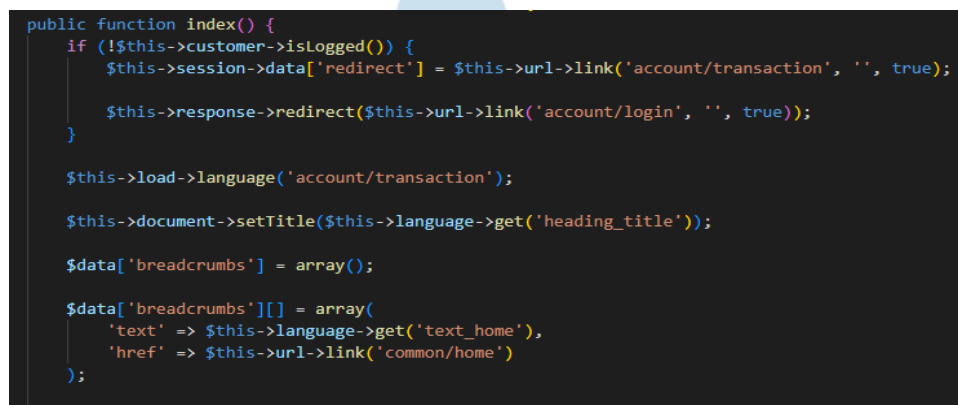
### 5.1.22 Implementasi Halaman Proses Pesanan

Pada gambar 5.43 menampilkan halaman proses pesanan dimana pelanggan akan menerima laporan akhir dari pesanan sebelum membuat pesanan. Pelanggan akan menerima laporan dari pesanan mulai dari produk, total harga dan alamat tujuan.

Pada gambar 5.44 merupakan potongan *code* yang menangani proses pesanan pelanggan.



Gambar 5.43 Implementasi Halaman Proses Pesanan



Gambar 5.44 Halaman Proses Pesanan *Controller*

## 5.2. Pengujian Fungsionalitas Perangkat Lunak

Dalam subbagian ini berisikan tabel pengujian terhadap fungsionalitas sistem pada ICKomp.

**Tabel 5.1 Pengujian terhadap fungsionalitas sistem pada ICKomp**

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
1	<b>Login Admin</b>	Pengujian Fungsi <i>login</i>	-Masukkan <i>username</i> dan <i>password</i> -Tekan Center atau tombol <i>login</i>	- <i>Username</i> admin “admin” - <i>Password</i> “admin”	Masuk ke dalam sistem dan menampilkan halaman beranda admin	Masuk ke dalam sistem dan menampilkan halaman beranda admin	Sudah Sesuai
2	<b>Halaman Beranda Admin</b>	Pengujian terhadap fungsi beranda admin	-Mengakses halaman beranda -Menampilkan halaman beranda admin -Berhasil <i>login</i> sebagai admin		Menampilkan halaman beranda admin	Menampilkan halaman beranda admin	Sudah Sesuai

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
3	<b>Pengaturan Kategori produk</b>	Pengujian terhadap fungsi Kategori produk	-Mengakses halaman kategori produk -Klik “Kategori” pada halaman pengaturan kategori		Menampilkan semua data kategori produk	Menampilkan semua data kategori produk yang tersedia	Sudah Sesuai
4	<b>Ubah kategori produk</b>	Pengujian terhadap fungsi ubah kategori produk	-Mengakses halaman kategori produk -Klik “Ubah” pada data kategori produk yang ingin diubah	-Nama kategori -Deskripsi kategori	Data kategori berhasil diubah	Data kategori berhasil diubah dan akan diperbarui pada halaman toko	Sudah Sesuai
5	<b>Tambah kategori produk</b>	Pengujian menambahkan kategori produk	-Mengakses halaman kategori produk -Klik “Tambah” pada data kategori produk yang ingin diubah	-Nama kategori -Deskripsi kategori	Data kategori berhasil ditambahkan	Data kategori berhasil ditambahkan dan akan muncul pada halaman toko	Sudah Sesuai



No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
			-Masukan nama kategori dan deskripsi kategori				
6	<b>Hapus Kategori</b>	Pengujian hapus kategori produk	-Mengakses halaman kategori produk -Pilih kategori produk yang ingin dihapus -Klik "Hapus"		Data kategori berhasil dihapus	Data kategori berhasil dihapus	Sudah sesuai
7	<b>Pengaturan Produk</b>	Pengujian terhadap fungsi pengaturan produk	-Mengakses halaman pengaturan produk -Klik "Produk" pada halaman pengaturan produk		Menampilkan semua data produk	Menampilkan semua data produk yang tersedia	Sudah Sesuai

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
8	<b>Ubah Produk</b>	Pengujian terhadap fungsi ubah produk	-Mengakses halaman pengaturan produk -Klik “Ubah” pada data kategori produk yang ingin diubah	-Nama produk -Deskripsi Produk	Data kategori berhasil diubah	Data produk berhasil diubah dan akan diperbarui pada halaman toko	Sudah Sesuai
9	<b>Tambah Produk</b>	Pengujian terhadap fungsi tambah produk	-Mengakses halaman pengaturan produk -Klik <i>icon</i> “Tambah” pada data produk -Masukan nama produk, deskripsi produk, gambar produk, harga, kuantiti	‘iPhone’, ‘HP iPhone dengan model terbaru’, ‘gambar’, ‘15.000.000’, ‘10’	Produk berhasil ditambahkan	Produk Iphone berhasil ditambahkan	Sudah Sesuai
10	<b>Hapus Produk</b>	Pengujian hapus produk	-Mengakses halaman pengaturan produk		Data produk berhasil dihapus	Data produk berhasil dihapus	Sudah sesuai

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
			-Pilih produk yang ingin dihapus -Klik "Hapus"				
11	<b>Statistik Umum</b>	Pengujian terhadap fungsi statistik umum	-Mengakses halaman Laporan dan Statistik -Klik halaman statistik umum		Menampilkan halaman statistik umum	Menampilkan halaman statistik umum setiap penjualan	Sudah Sesuai
12	<b>Pesanan Umum</b>	Pengujian terhadap fungsi pesanan umum	-Menampilkan halaman pesanan umum -Klik "Penjualan Toko" lalu klik "Pesanan Umum"		Menampilkan halaman pesanan umum	Menampilkan halaman pesanan umum setiap transaksi dari pembeli	Sudah Sesuai
13	<b>Rincian</b>	Pengujian	-Menampilkan		Menampilkan	Menampilkan	Sudah

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
	<b>Pelanggan</b>	terhadap fungsi rincian pelanggan umum	halaman rincian pelanggan -klik “Rincian Pelanggan”		halaman rincian pelanggan	halaman rincian pelanggan yang telah berhasil melakukan registrasi	Sesuai
14	<b>Pendaftaran Akun</b>	Pengujian terhadap fungsi pendaftaran akun	-Mengakses halaman <i>website</i> ICKomp -Mengakses halaman “Daftar Akun” -Masukan nama, email, nomor telepon, <i>password</i> , dan konfirmasi <i>password</i>	-“Rio Utomo”, “ <a href="mailto:rioutomo1@gmail.com">rioutomo1@gmail.com</a> ”, “0821234567”, “rio123”, “rio123”	Pendaftaran akun berhasil	Pendaftaran akun berhasil dan berhasil <i>login</i>	Sudah Sesuai
15	<b>Login Akun</b>	Pengujian terhadap fungsi	-Mengakses halaman <i>login</i>	‘rioutomo1@gmail.com’, ‘rio123’			

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
		<i>login</i> pengguna	-Masukan email dan <i>password</i> -Klik <i>login</i>				
16	<b>Halaman Beranda Pengguna</b>	Pengujian terhadap fungsi beranda pengguna	-Mengakses halaman beranda -Menampilkan halaman beranda pengguna -Berhasil <i>login</i> sebagai pengguna		Menampilkan halaman beranda pengguna	Menampilkan halaman beranda bagi pengguna untuk melihat produk yang tersedia dan melakukan transaksi	Sudah Sesuai
17	<b>Pengaturan Informasi Akun</b>	Pengujian terhadap fungsi informasi akun	-Mengakses halaman “Akun Saya” Klik “Informasi Akun”		Menampilkan halaman informasi akun	Menampilkan halaman informasi akun sesuai dengan data yang	Sudah Sesuai

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
						didaftarkan	
18	<b>Pengaturan Alamat</b>	Pengujian terhadap fungsi pengaturan alamat	-Mengakses halaman “Akun Saya” -Klik “Ubah Alamat” -Klik “Alamat Baru” -Memasukkan “nama”, “alamat”, “kota”, “provinsi”, “negara”, “kode pos”	“Rio Utomo”, “Jl. Parangtritis Km2. Desa tarudan kulon, Bangunharjo, Sewon”, “Bantul”, “Yogyakarta”, “55318”	Alamat berhasil disimpan	Alamat berhasil disimpan sebagai alamat pengiriman	Sudah Sesuai
19	<b>Pengaturan Password</b>	Pengujian terhadap fungsi pengaturan <i>password</i>	-Mengakses halaman “Akun Saya” -Klik “Ubah <i>Password</i> ” -Masukan “ <i>password</i> ”	“rio1234” dan “rio1234”	<i>Password</i> berhasil diubah	<i>Password</i> berhasil diubah	Sudah Sesuai

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
			baru” dan “konfirmasi <i>password</i> ”				
20	<b>Pilih Produk</b>	Pengujian fungsi pilih produk	-Mengakses halaman beranda -Pilih produk pilihan -Masukan kuantiti	“Iphone” dan “1”	Produk berhasil dipilih	Produk berhasil dipilih dan masuk pada keranjang belanja	Sudah Sesuai
21	<b>Keranjang Belanja</b>	Pengujian terhadap fungsi keranjang belanja	-Mengakses halaman keranjang belanja -Klik <i>icon</i> “keranjang”		Menampilkan halaman keranjang belanja	Menampilkan halaman keranjang belanja dengan produk yang telah dipilih	Sudah Sesuai
22	<b>Proses</b>	Pengujian	-Mengakses halaman	“Iphone”, “alamat	Pesanan berhasil di	Pesanan yang	Sudah

No	Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang digunakan	Hasil Yang didapat	Kesimpulan
	<b>Pesanan</b>	terhadap fungsi proses pesanan	keranjang belanja -Pilih produk yang akan diproses pesanannya -Klik “Proses Pesanan” -Memilih alamat pengiriman -Menambahkan catatan/komentar pada pengiriman -Pilih metode pembayaran” -Klik “Konfirmasi Pesanan”	pengiriman <i>default</i> ”, “ <i>packing</i> dengan aman”, “COD”	proses	dipilih berhasil diproses	Sesuai



### 5.3. Hasil Pengujian Terhadap Pengguna

Pada Tabel 5.2 berisikan tabel pengujian terhadap target pengguna sistem ICkomp yang dibangun. Pengujian aplikasi kepada pengguna menggunakan kuesioner. Responden yang ikut berpartisipasi dalam pengujian sebanyak 36 orang berperan sebagai pelanggan dan admin. Penilaian dilakukan dengan metode pilihan ganda dengan pilihan:

1. Sangat Setuju (SS)
2. Setuju (S)
3. Netral (N)
4. Tidak Setuju (TS)
5. Sangat Tidak Setuju (STS)

**Tabel 5.2. Pengujian Fungsional Sistem Terhadap Pengguna**

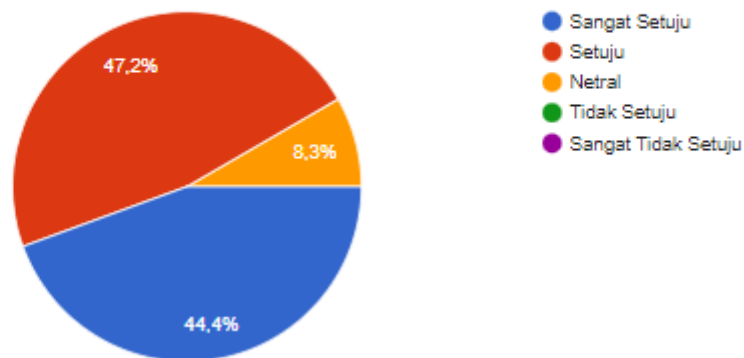
No	Pertanyaan	SS	S	N	TS	STS
1	Bagaimana tampilan antarmuka <i>website</i> ICkomp apakah memudahkan pengguna?	16	17	3	0	0
2	Bagaimana proses loading <i>website</i> ICkomp apakah sudah cepat?	9	20	7	0	0
3	Apakah <i>website</i> ICkomp mempermudah pelanggan dalam mencari katalog produk yang berada pada ICkomp?	12	21	3	0	0
4	Apakah <i>website</i> ICkomp memudahkan admin dalam proses pencatatan penjualan maupun pemesanan?	13	19	4	0	0

Berdasarkan hasil kuisioener yang diberikan pada pegawai dan pengguna ICkomp maka diperoleh hasil persentase pada setiap pertanyaan dengan menggunakan grafik pie yang terlihat seperti berikut:

1. Pertanyaan pertama

Bagaimana tampilan antarmuka website ICkomp apakah memudahkan pengguna?

36 jawaban



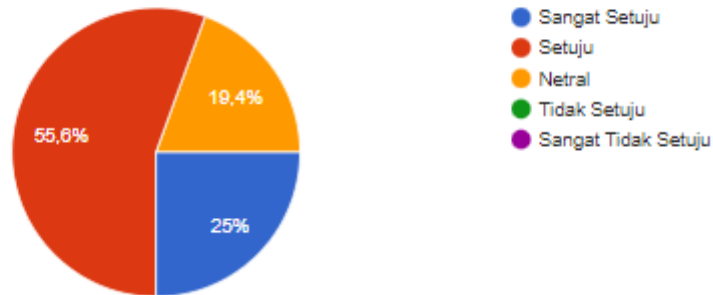
**Gambar 5.45 Pengujian Kemudahan Tampilan Antarmuka Bagi Pengguna**

Gambar 5.45 adalah gambar yang menampilkan grafik hasil kuesioner pada pertanyaan pertama mengenai “Bagaimana tampilan antarmuka *website* ICkomp apakah memudahkan pengguna?” dari pertanyaan tersebut didapatkan hasil enam belas (44,4%) menjawab sangat setuju, tujuh belas (47,2%) menjawab setuju, tiga (8,3%) menjawab netral, nol (0%) menjawab tidak setuju dan nol (0%) menjawab sangat tidak setuju, dari total 36 orang responden.

## 2. Pertanyaan kedua

Bagaimana proses loading website ICkomp apakah sudah cepat?

36 jawaban



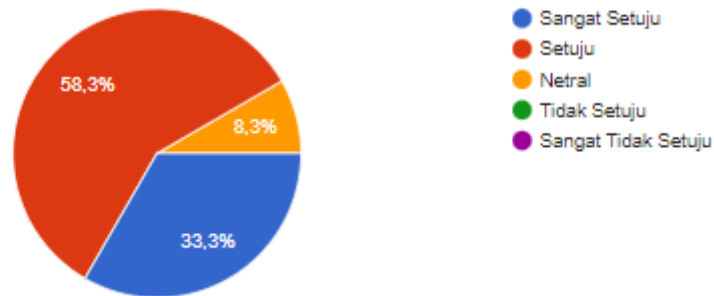
**Gambar 5.46** Pengujian proses Loading Website

Gambar 5.46 adalah gambar yang menampilkan grafik hasil kuesioner pada pertanyaan kedua mengenai “Bagaimana proses loading *website* ICkomp apakah sudah cepat?” dari pertanyaan tersebut didapatkan hasil sembilan (25%) menjawab sangat setuju, dua puluh (55,6%) menjawab setuju, tujuh (19,4%) menjawab netral, nol (0%) menjawab tidak setuju dan nol (0%) menjawab sangat tidak setuju, dari total 36 orang responden.

### 3. Pertanyaan ketiga

Apakah website ICkomp mempermudah pelanggan dalam mencari katalog produk yang berada pada ICkomp?

36 jawaban



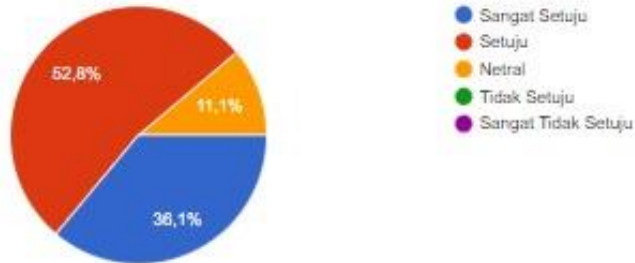
**Gambar 5.47 Pengujian Kemudahan Mencari Katalog Produk**

Gambar 5.47 adalah gambar yang menampilkan grafik hasil kuesioner pada pertanyaan ketiga mengenai “Apakah *website* ICkomp mempermudah pelanggan dalam mencari katalog produk yang berada pada ICkomp?” dari pertanyaan tersebut didapatkan hasil dua belas (33,3%) menjawab sangat setuju, dua puluh satu (58,3%) menjawab setuju, tiga (8,5%) menjawab netral, nol (0%) menjawab tidak setuju dan nol (0%) menjawab sangat tidak setuju, dari total 36 orang responden.

#### 4. Pertanyaan keempat

Apakah website ICkomp memudahkan admin dalam proses pencatatan penjualan maupun pemesanan?

36 jawaban



**Gambar 5.48 Pengujian Kemudahan Mencari Katalog Produk**

Gambar 5.48 adalah gambar yang menampilkan grafik hasil kuesioner pada pertanyaan keempat mengenai “Apakah *website* ICkomp memudahkan admin dalam proses pencatatan penjualan maupun pemesanan?” dari pertanyaan tersebut didapatkan hasil tiga belas (36,1%) menjawab sangat setuju, sembilan belas (52,8%) menjawab setuju, empat (11,1%) menjawab netral, nol (0%) menjawab tidak setuju dan nol (0%) menjawab sangat tidak setuju, dari total 36 orang responden.