

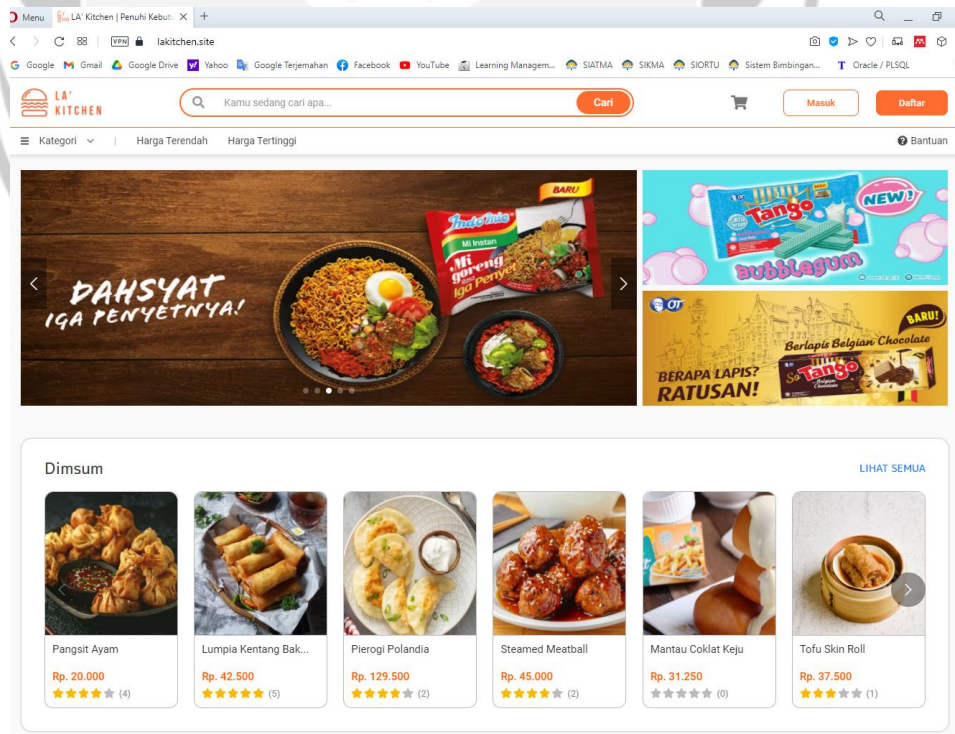
# BAB V. IMPLEMENTASI DAN PENGUJIAN SISTEM

## 5.1. Implementasi Sistem Implementasi Antarmuka

Pada subbagian ini berisikan pembahasan terkait alur jalannya sistem informasi yang telah jadi. Berikut ini merupakan alur jalannya sistem informasi yang disertai dan didukung dengan *screenshot* dan penjelasan dari kode - kode yang bersangkutan.

### 1) Halaman *Home*

Halaman ini digunakan sebagai halaman utama dan pertama ketika pengguna mengakses *website* LA' Kitchen. Seperti yang terlihat pada Gambar 5.1. dan Kode 5.1. halaman ini berisi berbagai macam komponen, seperti *slider* yang berisi produk berdasarkan sub kategori, *banner slider* yang digunakan untuk iklan, *header* yang berisi kategori produk, *filter*, *search*, dan komponen pendukung lainnya. Sistem pada sisi server akan mengeluarkan data yang diminta pada halaman *home*, seperti mengambil produk berdasarkan kategori atau sub kategori.



Gambar 5.1. Potongan halaman *home*.

```

public ResponseTemplate getBySubCategoryLimit(Integer
subCategoryId, Integer limit) {
    BasicResult result = this
        .validationGetBySubCategory(subCategoryId);

    if (result.getResult()) {
        ArrayList<Product> products = productRepository
            .findBySubCategoryLimit(subCategoryId, limit);
        ArrayList<ProductGeneralDTO> productDTO =
            new ArrayList<>();
        ProductSubCategory subCategory =
            productSubCategoryRepository
                .findFirstById(subCategoryId);

        products.forEach((val) -> {
            ProductGeneralDTO dto = productMapper
                .setToFormatGeneralDTO(val);
            productDTO.add(dto);
        });

        return new ResponseTemplate(200, "OK",
            new GetBySubCategory(subCategory.getName(),
                productDTO), null, null
        );
    }
}

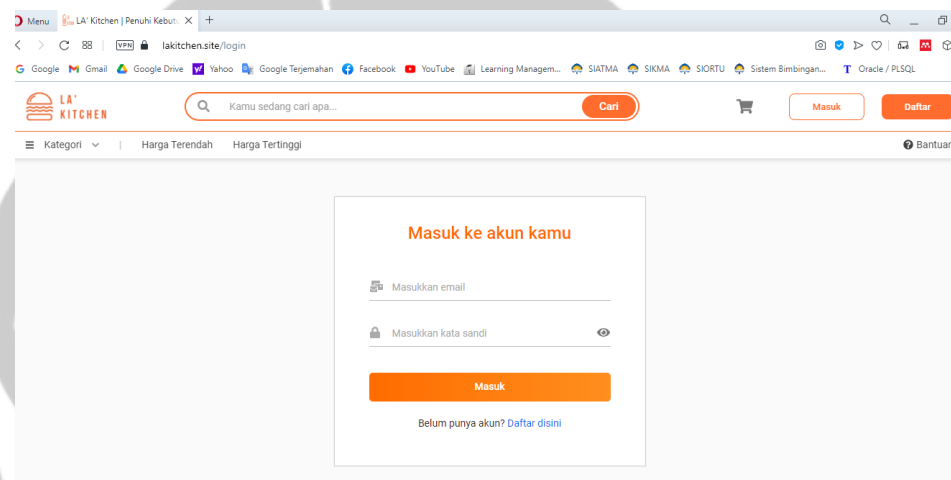
```

**Kode 5.1. Potongan kode *home*.**

2) Halaman *Login*

Halaman ini digunakan untuk melakukan autentikasi terhadap pengguna terutama yang memiliki *role user*, sedangkan untuk autentikasi pengguna seperti *admin* atau pegawai berada di halaman yang berbeda. Sekalipun berada di halaman yang berbeda, pengecekan para pengguna sistem ini menggunakan konsep dan cara yang sama dalam melakukan autentikasi dan otorisasi, yaitu menggunakan *Json Web Token (JWT)*. Seperti yang terlihat pada Gambar 5.2. sistem meminta pengguna untuk memasukkan alamat *email* dan kata sandi, kemudian *inputan* ini akan diproses pada sistem *backend* seperti yang terlihat pada Kode 5.2. Pertama sistem akan melakukan validasi terlebih dahulu, apakah *email* dan kata sandi sudah dimasukkan oleh pengguna, setelah itu sistem akan melakukan autentikasi berdasarkan basis data yang terpusat. Jika autentikasi berhasil, maka sistem *backend* atau *server* akan mengembalikan data kepada sistem *frontend* berupa data pengguna

beserta token JWT yang telah dilakukan *encoding*. Token ini berfungsi sebagai kunci masuk untuk mengakses fitur - fitur LA' Kitchen nantinya. Tanpa adanya token ini, maka sistem menganggap bahwa pengguna berusaha mengakses sebuah fungsi secara paksa tanpa mengikuti prosedur sistem. Jika terjadi hal yang demikian, maka sistem *backend* akan mengembalikan kode *403* yang berarti *forbidden* atau dilarang mengakses. Hal ini dilakukan untuk memastikan setiap pengguna telah terdaftar pada sistem, dan memiliki otorisasi terhadap fungsi sistem berdasarkan *role* yang telah ditetapkan.



**Gambar 5.2. Potongan halaman *login*.**

```
public ResponseTemplate login(LoginRequest request) {
    BasicResult result = this.validationLogin(request);

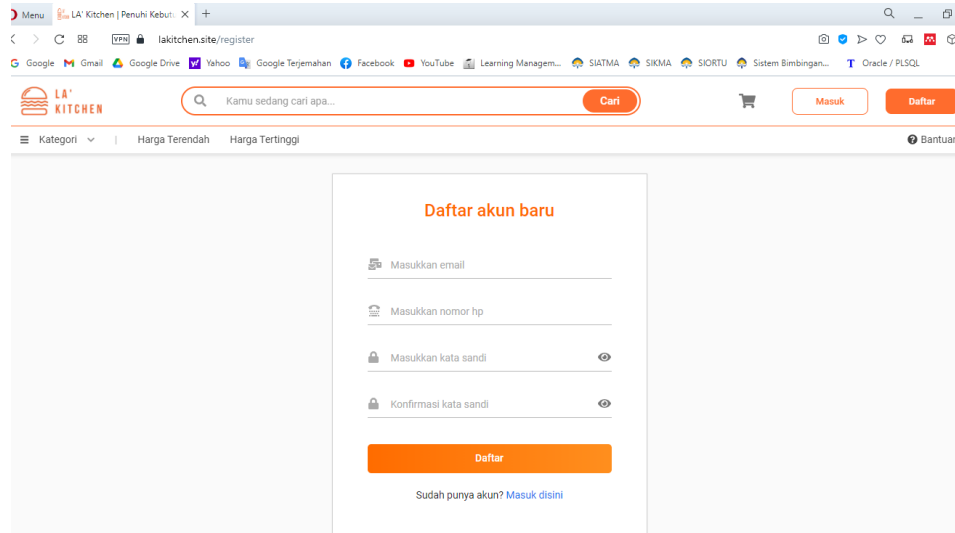
    if (result.getResult()) {
        Authentication authentication =
            authenticationManager.authenticate(
                new UsernamePasswordAuthenticationToken(
                    request.getEmail(),
                    request.getPassword()
                )
            );
        SecurityContextHolder.getContext()
            .setAuthentication(authentication);
        String jwt = tokenProvider
            .generateToken(authentication);
        User user = userRepository
            .findFirstByEmail(request.getEmail());
        UserDTO userDTO = userMapper.mapToUserDTO(user);
    }
}
```

```
        return new ResponseTemplate(200, "OK",  
            new Login(jwt, userDTO), null, null);  
    }  
}
```

### **Kode 5.2. Potongan kode *login*.**

#### 3) Halaman *Register*

Halaman ini digunakan untuk melakukan pendaftaran akun baru. Seperti yang terlihat pada Gambar 5.3. sistem meminta pengguna untuk memasukkan alamat *email*, nomor telepon, dan kata sandi. Alamat *email* sangat penting dalam melakukan pendaftaran, karena sistem akan mengirimkan verifikasi kepada *email* pendaftar untuk mengaktifkan akun. Hal ini dilakukan dengan tujuan memastikan bahwa alamat *email* pendaftar merupakan alamat email yang aktif, sehingga ketika melakukan transaksi, sistem dapat mengirimkan pesan kepada *email* pengguna. Tujuan lainnya adalah untuk membatasi pengguna melakukan perubahan alamat *email* yang palsu, maka dari itu sejak awal pendaftaran sistem langsung melakukan verifikasi dan menetakannya sebagai alamat *email* yang aktif. Seperti yang terlihat pada Kode 5.3. sistem *backend* akan memproses parameter yang dimasukkan oleh sistem *frontend*, yang kemudian menyimpannya pada objek User. Setelah melakukan penyimpanan pada objek, sistem tidak langsung menyimpannya pada *database*, melainkan sistem melakukan pengiriman verifikasi *email* kepada pendaftar, dan memastikan bahwa verifikasi *email* tersebut telah berhasil dikirim. Setelah itu, sistem menyimpan data pendaftar pada *database*. Jika pengiriman verifikasi *email* tidak berhasil, maka sistem tidak akan menyimpan data pendaftar pada *database* dan menganggap bahwa itu merupakan *internal server error*. Hal ini dilakukan untuk memastikan setiap pendaftar yang tersimpan pada *database* telah menerima pesan verifikasi pada *email* yang didaftarkan.



**Gambar 5.3. Potongan halaman register.**

```

public ResponseTemplate register(RegisterRequest request) {
    BasicResult result = this.validationRegister(request);

    if (result.getResult()) {
        User user = new User();

        String userMail = request.getEmail();
        String userPass = this
            .BCryptEncoder(request.getPassword());
        user.setUserStatus(this.unActiveStatus());
        user.setUserRole(this.roleUser());
        user.setEmail(userMail);
        user.setPassword(userPass);
        user.setPhoneNumber(request.getPhoneNumber());
        user.setCreatedAt(FUNC.getCurrentTimestamp());

        this.sendActivationMail(userMail, userPass);
        userRepository.save(user);

        return new ResponseTemplate(200, "OK", null,
            null, null);
    }

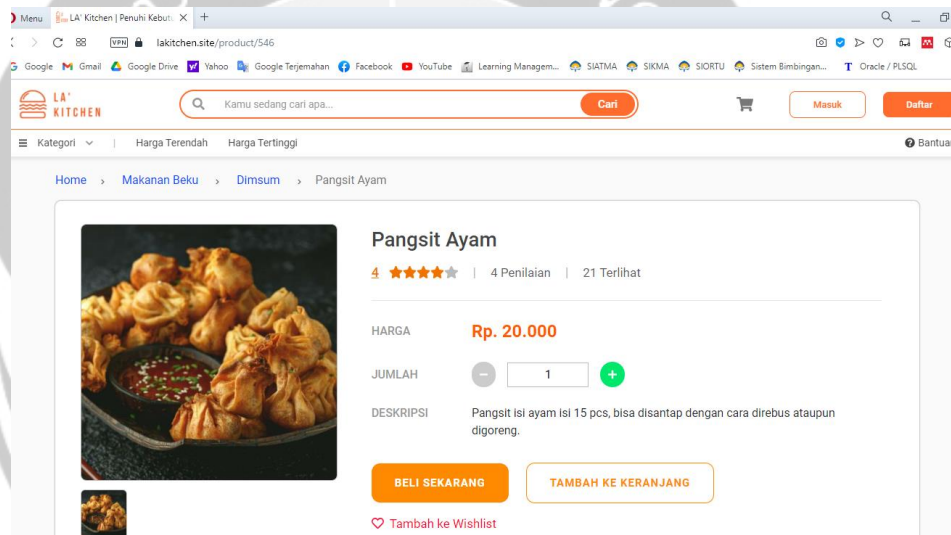
    return new ResponseTemplate(result.getCode(),
        result.getStatus(), null, null, result.getError());
}

```

**Kode 5.3. Potongan kode register.**

#### 4) Halaman Detail Produk

Halaman ini digunakan untuk mendeskripsikan seluruh informasi yang berhubungan dengan produk yang dilihat. Seperti yang terlihat pada Gambar 5.4. pengguna dapat mengetahui informasi produk seperti nama produk, harga, *rating*, ulasan, foto produk, dan yang lainnya. Pada Kode 5.4. terlihat bahwa ketika sistem *frontend* telah mendapatkan data yang diperlukan, maka sistem melakukan *request* lagi untuk menambah jumlah terlihat pada produk. Disaat yang bersamaan, sistem juga mengatur fitur *zoom* pada foto produk. Melalui fitur *zoom* ini, pengguna dapat melihat foto produk dengan lebih jelas. Pada halaman ini juga, pengguna dapat menyimpan produk ke dalam *wishlist* ataupun *cart*.



**Gambar 5.4. Potongan halaman detail produk.**

```

async getProductDetail() {
  this.loader = true;
  const { code } = await this
    .$func.promiseAPI(this.getById, {
      productId: this.paramId,
    });

  this.loader = false;
  if (code >= 200 && code < 300) {
    if (!this.isActiveProduct()) {
      this.$func.popupInfoWithBack(
        'Mohon maaf produk ini dalam status nonaktif',
        this.$router,
      );
    }

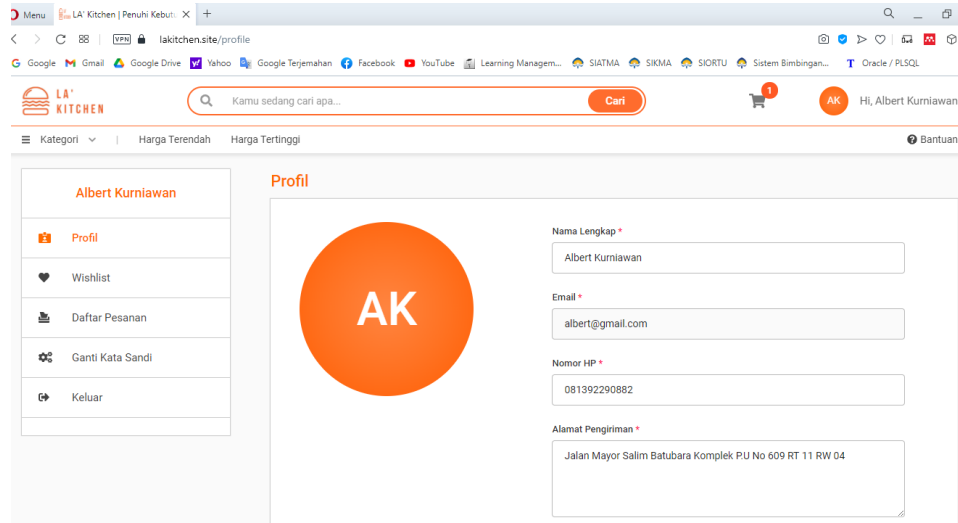
    this.path = this.productDetail.path;
    this.product = this.productDetail.product;
    [this.imgBinding] = this.product.photo_links;
    this.imgBinding = this.imgBinding.link;
    setTimeout(() => {
      this.imgZoom();
    }, 1);
    this.postIncrementSeen();
  } else {
    this.$func.popupConnectionError();
  }
}

```

**Kode 5.4. Potongan kode detail produk.**

#### 5) Halaman Profil

Halaman ini digunakan untuk melihat dan melengkapi biodata diri pengguna sistem. Seperti yang terlihat pada Gambar 5.5. bahwa sistem meminta beberapa data diri pengguna. Hal ini diperlukan sebagai data pengguna yang melakukan transaksi, sehingga penjual dapat mengetahui siapa yang melakukan transaksi, alamat pembeli, dan yang lainnya. Pada Kode 5.5. terlihat bahwa sistem *frontend* tidak melakukan *request* kepada sistem *backend* untuk mendapatkan data pengguna, melainkan cukup mengambil data tersebut dari *cookies* yang telah tersimpan.



**Gambar 5.5. Potongan halaman profil.**

```

setForm() {
  const user = this.$cookies.get('user');
  this.fillCity({ target: { value: user.province } });

  this.form = {
    name: user.name === null ? '' : user.name,
    email: user.email,
    phoneNumber: user.phoneNumber,
    address: user.address === null ? '' : user.address,
    province: user.province === null ? '' : user.province,
    city: user.city === null ? '' : user.city,
  };
}

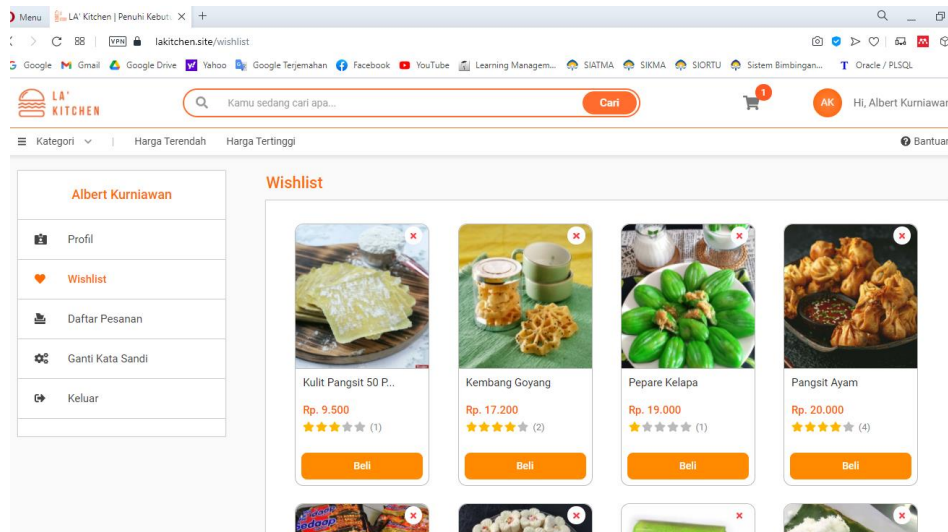
```

**Kode 5.5. Potongan kode profil.**

#### 6) Halaman *Wishlist*

Halaman ini digunakan untuk menyimpan semua produk yang disukai oleh pengguna. Seperti yang terlihat pada Gambar 5.6. pengguna dapat melihat, menghapus, bahkan langsung membeli produk yang telah disimpan pada *wishlist*. Terlihat pada Kode 5.6. bahwa sistem akan mengambil seluruh produk yang tersimpan pada akun pengguna tersebut, yang kemudian diolah dan ditampilkan kepada pengguna.





**Gambar 5.6. Potongan halaman wishlist.**

```

public ResponseTemplate getAll(Integer userId) {
    BasicResult result = this.validationGetAll(userId);

    if (result.getResult()) {
        ArrayList<Wishlist> wishlists = wishlistRepository
            .findByUser_Id(userId);
        ArrayList<ProductGeneralDTO> dto = new ArrayList<>();

        wishlists.forEach((val) -> {
            dto.add(productMapper
                .setToFormatGeneralDTO(val.getProduct()));
        });

        return new ResponseTemplate(200, "OK",
            new GetByUserId(dto), null, null);
    }
}

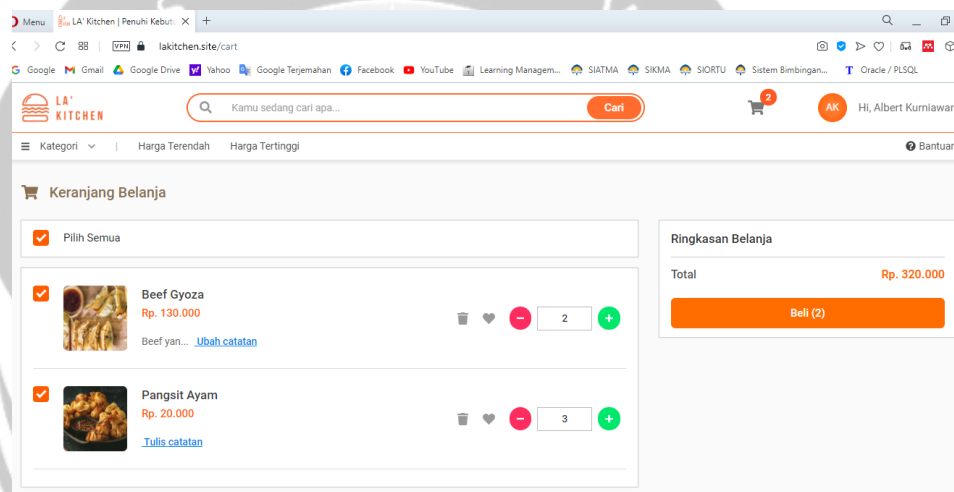
```

**Kode 5.6. Potongan kode wishlist.**

## 7) Halaman Cart

Halaman *cart* digunakan seperti layaknya keranjang di dunia nyata yaitu untuk menampung atau menyimpan produk - produk yang akan atau ingin dibeli. Seperti yang terlihat pada Gambar 5.7. dan Kode 5.7. bahwa sistem menampilkan produk - produk dan jumlah produk yang ada di keranjang pelanggan. Pelanggan dapat melakukan manipulasi terhadap produk yang ada di keranjang seperti menambah atau mengurangi jumlah beli, menambahkan catatan pada toko, menghapus produk dari

keranjang, memilah produk yang akan dibeli, hingga menyimpan produk tersebut ke dalam *wishlist* sebagai produk yang disukai dan ingin disimpan. Fitur dan halaman ini hanya dapat diakses oleh pengguna yang telah dilakukan autentikasi dan memiliki otorisasi sebagai *role user*. Sistem juga secara otomatis menghitung ringkasan belanja pelanggan yang ditampilkan dalam bentuk rupiah. Untuk melakukan *checkout* atau pembelian produk, pelanggan harus memilih minimal satu produk. Jika pelanggan tidak memilih satu produk pun, maka tombol beli pada halaman ini secara otomatis akan menjadi *disable*. Hal ini dilakukan untuk memastikan bahwa pengguna telah memilih minimal satu produk yang akan dibawa ke dalam pembayaran.



**Gambar 5.7. Potongan halaman cart.**

```

async getTotalCart() {
  await this.$func.promiseAPI(this.countCart, {
    userId: this.$cookies.get('user').id,
  });
},

async getUserCart() {
  this.loader = true;
  const { code } = await this.$func.promiseAPI(this.getAll, {
    userId: this.$cookies.get('user').id,
  });
  this.loader = false;
}

```

```

if (code >= 200 && code < 300) {
    this.products = this.userCart.products;
    this.selectedProduct = this.products;
} else {
    this.products = [];
    this.$func.popupConnectionError();
}
}

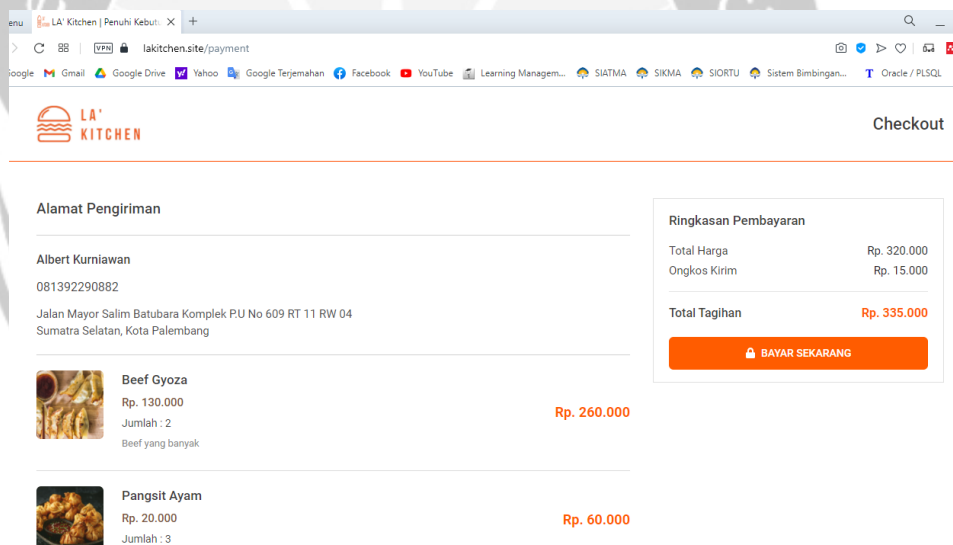
```

#### **Kode 5.7. Potongan kode *cart*.**

#### 8) Halaman *Checkout*

Halaman ini merupakan halaman terakhir untuk mengkonfirmasi transaksi yang akan dilakukan. Seperti yang terlihat pada Gambar 5.8. bahwa sistem akan menampilkan data pengguna sebagai alamat pengiriman, produk - produk yang dibeli beserta datanya seperti jumlah beli, catatan pada toko, ataupun sub total harga, hingga ringkasan pembayaran dan total tagihan yang perlu dibayarkan. Seperti yang terlihat pada Kode 5.8. bahwa pertama sistem akan mengecek rute asal sebelum masuk ke halaman *checkout*. Hal ini perlu dilakukan untuk memastikan bahwa pengguna yang masuk ke halaman *checkout* berasal dari halaman *cart*, karena produk - produk yang ingin dibeli hanya dapat dilakukan pada halaman *cart*. Kedua, sistem akan mengecek autentikasi dan mengambil data produk yang dibawa dari keranjang. Jika sistem tidak menemukan data produk dari keranjang, maka sistem akan mengembalikan pengguna ke halaman *cart*. Hal itu terjadi karena sistem menganggap bahwa pengguna tidak memilih satupun produk dari keranjang dan berusaha masuk ke halaman *checkout* tanpa melalui validasi yang terjadi pada halaman *cart*. Ketiga, sistem akan melakukan pengecekan terhadap data profil pengguna. Jika pengguna belum melengkapi data diri, maka sistem akan menampilkan pesan dan mengembalikan pengguna ke halaman profil untuk melengkapi data diri. Data diri dan yang lainnya sangat diperlukan dalam melakukan transaksi, maka dari itu sistem perlu memastikan semuanya telah melewati

prosedur yang ada. Jika pelanggan meninggalkan halaman *checkout*, maka sistem akan menghapus data keranjang yang dibawa ke halaman *checkout* pada *storage* pengguna, dengan tujuan mengembalikan segala sesuatu seperti semula sebelum pelanggan masuk ke halaman *checkout*. Setelah semuanya telah sesuai dengan prosedur dan melewati validasi yang ada, pelanggan dapat melakukan konfirmasi pembayaran. Setelah pelanggan melakukan konfirmasi pembayaran, maka sistem akan mengirimkan pesan pada *email* pengguna dan penjual terkait transaksi yang dilakukan. Tujuan dikirimkan kepada pengguna, agar pengguna mengetahui nomor rekening dan total tagihan yang perlu dibayarkan. Sedangkan kepada penjual, pesan *email* tersebut sebagai *reminder* atau pengingat bahwa ada transaksi yang telah terjadi, sehingga penjual dapat melihat produk - produk yang dibeli oleh pelanggan, alamat pengiriman, serta yang lainnya.



**Gambar 5.8. Potongan halaman *checkout*.**

```

beforeRouteLeave(to, from, next) {
  localStorage.removeItem('cart');
  next();
},

beforeRouteEnter(to, from, next) {
  if (from.name === 'Cart' && to.name === 'Payment') {
    next();
  } else {
    window.location.href = '/cart';
  }
},

created() {
  this.$func.isLoggedIn(this.$cookies, true, this.$router);
  this.cart = JSON.parse(localStorage.getItem('cart'));
  this.user = this.$cookies.get('user');

  if (!this.cart) {
    this.$router.go(-1);
  }

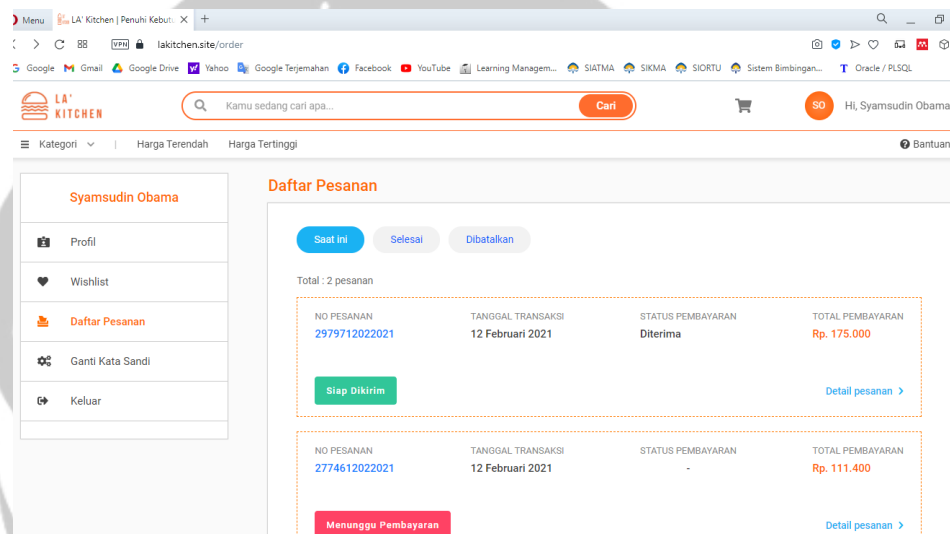
  if (!this.isProfileComplete()) {
    this.$func.popupErrorRedirect(
      'Lengkapi profil terlebih dahulu',
      'Ok',
      this.$router,
      '/profile',
    );
  }
}
}

```

**Kode 5.8. Potongan kode *checkout*.**

## 9) Halaman Daftar Pesanan

Halaman ini digunakan untuk melihat semua pesanan atau transaksi yang telah dilakukan. Seperti yang terlihat pada Gambar 5.9. bahwa sistem menampilkan data - data pesanan, seperti status pesanan, total pembayaran, ataupun tanggal transaksi. Seperti yang terlihat pada Kode 5.9. bahwa sistem *backend* mengambil data tersebut dari beberapa tabel, seperti tabel *order* dan tabel *payment* yang kemudian diberikan kepada sistem *frontend*. Pengguna juga dapat melakukan *filter* untuk melihat transaksi yang sedang aktif, selesai, atau dibatalkan.



Gambar 5.9. Potongan halaman daftar pesanan.

```
public ResponseTemplate getAllOrder(Integer userId) {
    BasicResult result = this.validationGetAllOrder(userId);

    if (result.getResult()) {
        ArrayList<Order> orders = orderRepository
            .findByUser_IdOrderByCreatedAtDesc(userId);

        ArrayList<OrderGeneralDTO> dtos = new ArrayList<>();
        orders.forEach((val) -> {
            Payment payment = paymentRepository
                .findFirstByOrder_OrderNumber(
                    val.getOrderNumber()
                );
            dtos.add(orderMapper
                .mapToOrderGeneralDTO(
                    val, payment, val.getOrderStatus()
                )
            );
        });
    }
}
```

```

    );
  });
  return new ResponseTemplate(200, "OK",
    new GetByUserId(dtos), null, null);
}

return new ResponseTemplate(result.getCode(),
  result.getStatus(), null, null, result.getError());
}

```

### Kode 5.9. Potongan kode daftar pesanan.

#### 10) Halaman Detail Pesanan

Halaman ini digunakan untuk melihat detail dari pesanan yang telah dilakukan. Seperti yang terlihat pada Gambar 5.10. bahwa selain menampilkan data transaksi, sistem juga menampilkan detail produk yang dibeli dalam pesanan tersebut. Seperti yang terlihat pada Kode 5.10. bahwa selain mengambil data dari tabel *order* dan *payment*, sistem *backend* juga mengambil data yang berasal dari tabel *order\_detail* dan *product* yang kemudian diberikan kepada sistem *frontend*. Pengguna dapat melihat data produk yang dibeli dalam bentuk tabel, seperti nama produk, harga, jumlah beli, sub total, dan catatan yang ditulis untuk toko.

The screenshot shows the 'Detail Pesanan' page for user Syamsudin Obama. The page layout includes a user profile sidebar on the left and a main content area with order details and a product table.

NO PESANAN	TANGGAL TRANSAKSI	TOTAL PEMBAYARAN	STATUS PESANAN
2979712022021	12 Februari 2021	Rp. 175.000	Siap Dikirim

PRODUK	HARGA	JUMLAH	SUBTOTAL	CATATAN
Slomay Ikan x Udadang	Rp. 60.000	1	Rp. 60.000	
Pangsit Ayam	Rp. 20.000	5	Rp. 100.000	
Ongkos Kirim	Rp. 15.000	-	-	

Gambar 5.10. Potongan halaman detail pesanan.

```

public ResponseTemplate getDetailByOrderNumber(String
orderNumber) {
    BasicResult result = this
        .validationGetDetailByOrderNumber (orderNumber);

    if (result.getResult()) {
        Order order = orderRepository
            .findFirstByOrderNumber (orderNumber);
        Payment payment = paymentRepository
            .findFirstByOrder_OrderNumber (orderNumber);

        ArrayList<OrderDetail> orderDetails =
            orderDetailRepository
                .findByOrder_OrderNumber (orderNumber);

        ArrayList<ProductOrderDTO> productOrderDTOS = this
            .helperMapToProductOrder (orderDetails);

        return new ResponseTemplate (200, "OK",
            New GetDetailByNumber (orderMapper
                .mapToOrderDetailDTO (order,
                    payment, order.getOrderStatus (),
                    productOrderDTOS)),
            null, null);
    }

    return new ResponseTemplate (result.getCode (),
        result.getStatus (), null, null, result.getError ()
    );
}

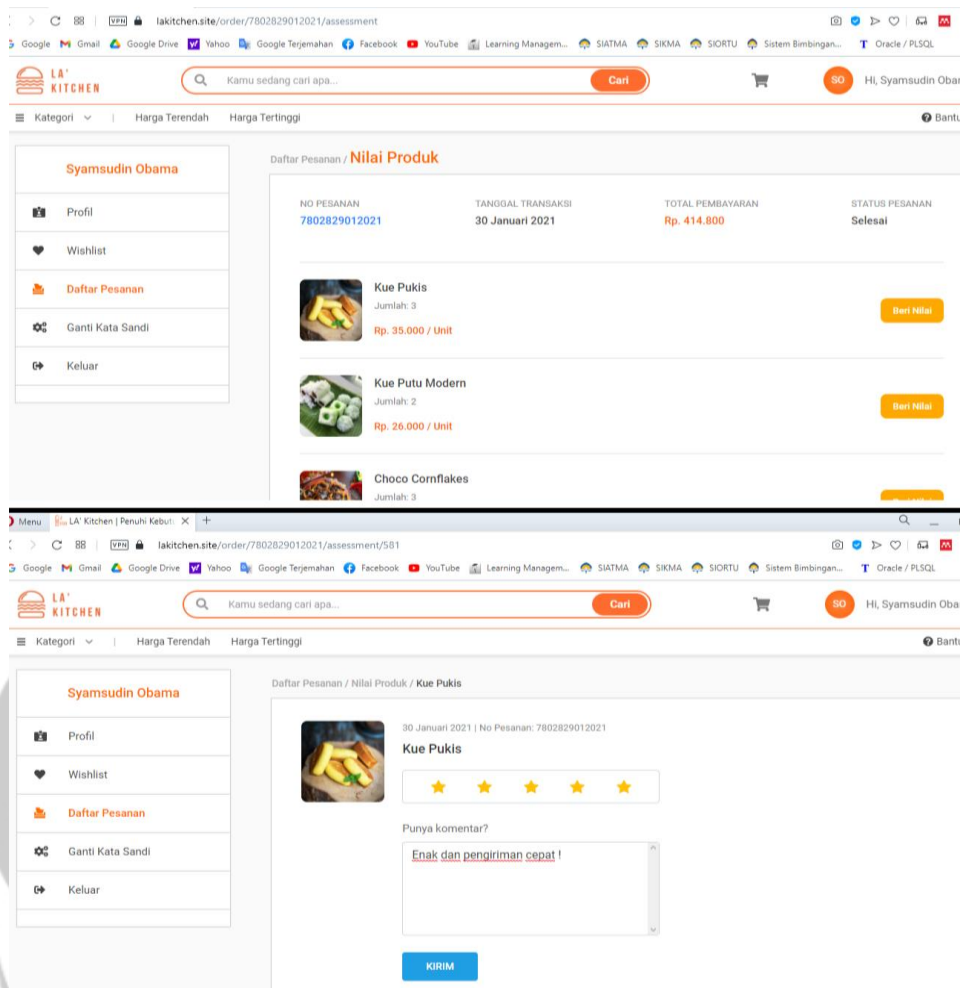
```

**Kode 5.10. Potongan kode detail pesanan.**

11) Halaman Penilaian Produk

Halaman ini digunakan untuk melakukan penilaian terhadap suatu produk. Pengguna dapat melakukan penilaian produk jika transaksi yang dilakukan telah selesai. Seperti yang terlihat pada Gambar 5.11. bahwa sistem meminta penilaian produk berupa jumlah bintang dan komentar, yang akan diolah dan dijadikan sebagai *rating* sebuah produk. Pada Kode 5.11. terlihat bahwa sistem akan mengecek apakah transaksi tersebut telah selesai. Jika telah selesai, maka sistem akan melakukan validasi terhadap data dan menyimpannya pada *database*.





**Gambar 5.11. Potongan halaman penilaian produk.**

```

public ResponseTemplate postAssessment (
AssessmentPostRequest request) {
    BasicResult result = this
        .validationPostAssessment (request);

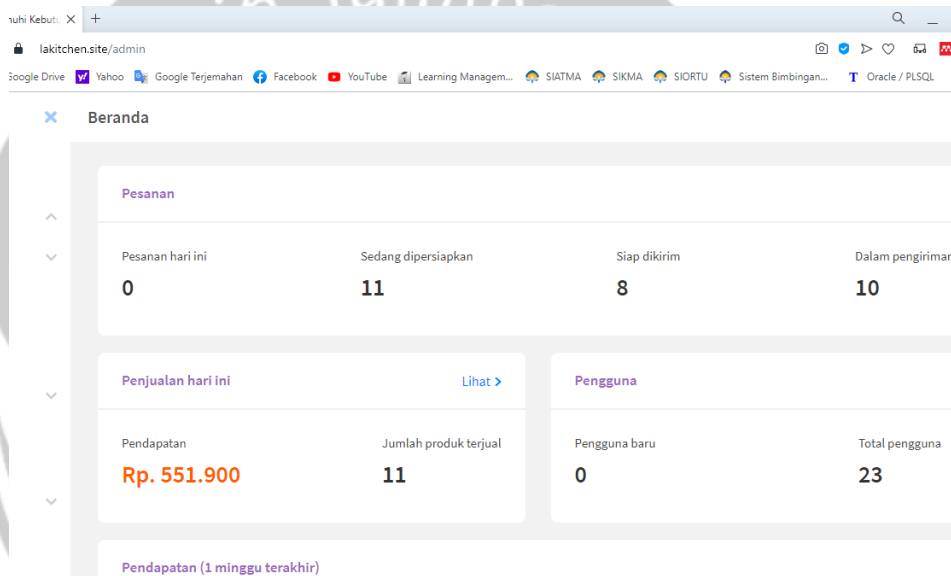
    if (result.getResult()) {
        ProductAssessment assessment = new ProductAssessment ();
        OrderDetail orderDetail = orderDetailRepository
            .findFirstByOrder_OrderNumberAndProduct_Id (
                request.getOrderNumber (), request.getProductId ());
        assessment.setComment (request.getComment ());
        assessment.setCreatedAt (FUNC.getCurrentTimestamp ());
        assessment.setRate (Double.valueOf (request.getRate ());
        assessment.setUser (userRepository
            .findFirstById (request.getUserId ());
        assessment.setProduct (productRepository
            .findFirstById (request.getProductId ());
        orderDetail.setIsAssessment (1);
    }
}

```

**Kode 5.11. Potongan kode penilaian produk.**

## 12) Halaman Beranda *Admin*

Halaman ini merupakan halaman pertama yang akan dijumpai oleh *admin* ketika berhasil melakukan *login*. Seperti yang terlihat pada Gambar 5.12. dan Kode 5.12. bahwa sistem menampilkan berbagai macam informasi penting, antara lain jumlah pesanan hari ini, jumlah pesanan berdasarkan status, penjualan dan pendapatan hari ini, jumlah pengguna baru, serta pendapatan selama satu minggu terakhir. Pendapatan satu minggu terakhir ditampilkan dalam bentuk grafik garis, sehingga *admin* atau penjual dapat melihat pertumbuhan ekonomi yang dialaminya.



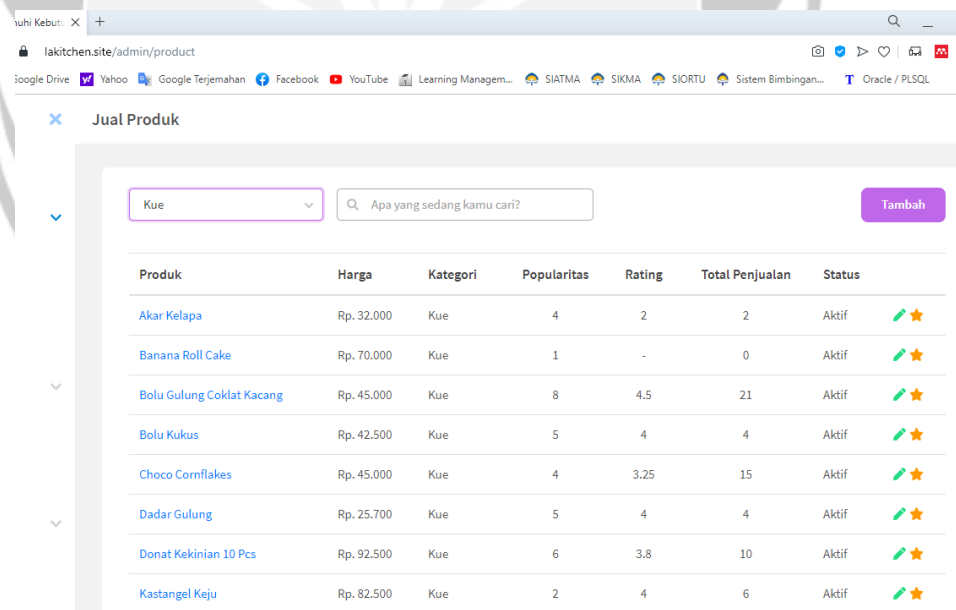
**Gambar 5.12. Potongan halaman beranda *admin*.**

```
async getAllData() {
  const { code } = await this.$func.promiseAPI(this.getAll)
  if (code >= 200 && code < 300) {
    const { orders, sales, others } = this.dashboardData;
    this.orders = orders;
    this.sales = sales;
    this.others = others;
    this.dataGraphFormat(this.dashboardData.weeklyReport,
      'date', 'income');
  }
}
```

**Kode 5.12. Potongan kode beranda *admin*.**

### 13) Halaman Pengelolaan Produk

Halaman ini digunakan oleh *admin* untuk melakukan pengelolaan terhadap produk yang dijual. Pertama, seperti yang terlihat pada Gambar 5.13. bahwa sistem secara *default* akan menampilkan seluruh data produk yang terdaftar dalam bentuk *paging*. Selain menampilkan informasi data produk, sistem juga mengikutsertakan laporan produk seperti popularitas, *rating*, dan total penjualan dari produk tersebut. Kedua, pengguna dapat melakukan *filter* berdasarkan kategori produk. Ketiga, pengguna juga dapat melakukan pencarian berdasarkan nama produk. Selain itu, dari halaman ini pengguna juga dapat memilih untuk melakukan penambahan, pengubahan, ataupun penonaktifan produk. Dari halaman ini juga, pengguna dapat memilih untuk melihat detail penilaian produk yang telah dilakukan oleh pembeli. Pada Kode 5.13. terlihat bahwa *paging* yang digunakan untuk menampilkan produk adalah sebanyak sepuluh data per *page*, dan dilakukan *sorting* secara *ascending* yang berarti pengurutan berdasarkan abjad dari a - z.



The screenshot shows a web browser window with the URL 'lakitchen.site/admin/product'. The page title is 'Jual Produk'. At the top, there is a search bar with the placeholder text 'Apa yang sedang kamu cari?' and a 'Tambah' button. Below the search bar is a table with the following columns: Produk, Harga, Kategori, Popularitas, Rating, Total Penjualan, and Status. The table contains eight rows of product data.

Produk	Harga	Kategori	Popularitas	Rating	Total Penjualan	Status
Akar Kelapa	Rp. 32.000	Kue	4	2	2	Aktif
Banana Roll Cake	Rp. 70.000	Kue	1	-	0	Aktif
Bolu Gulung Coklat Kacang	Rp. 45.000	Kue	8	4.5	21	Aktif
Bolu Kukus	Rp. 42.500	Kue	5	4	4	Aktif
Choco Cornflakes	Rp. 45.000	Kue	4	3.25	15	Aktif
Dadar Gulung	Rp. 25.700	Kue	5	4	4	Aktif
Donat Kekinian 10 Pcs	Rp. 92.500	Kue	6	3.8	10	Aktif
Kastangel Keju	Rp. 82.500	Kue	2	4	6	Aktif

**Gambar 5.13. Potongan halaman pengelolaan produk.**

```

public ResponseTemplate getAll(Integer page) {
    Page<Product> products = productRepository
        .findAll(PageRequest.of((page-1), 10,
            Sort.by("name").ascending()));
    ArrayList<ProductAdminDTO> productDTOS =
        new ArrayList<>();

    products.getContent().forEach((val) -> {
        productDTOS.add(this.helperMapToProductAdminDTO(val));
    });
    PageableDTO pageableDTO = FUNC
        .mapToPageableDTO(products);
}

```

### Kode 5.13. Potongan kode tampil produk.

#### 14) Halaman Tambah Produk

Halaman ini berisi *form* yang dapat digunakan oleh *admin* untuk melakukan penambahan ataupun pengubahan pada produk. Seperti yang terlihat pada Gambar 5.14. sistem akan meminta beberapa data yang perlu dimasukkan oleh pengguna. Setiap *inputan* yang ada pada *form* dapat memiliki format dan tipe data yang berbeda. Pada Kode 5.14. sistem akan melakukan validasi terhadap data yang dimasukkan oleh pengguna. Jika semuanya telah sesuai prosedur, maka sistem akan menyimpan atau memperbarui data produk tersebut pada *database*.

The screenshot shows a web browser window with the URL 'lakitchen.site/admin/product/form'. The page title is 'Jual Produk > Tambah Produk'. The form contains the following elements:

- Nama produk:** A text input field with the placeholder 'Masukkan nama produk'.
- Deskripsi produk:** A text area with the placeholder 'Tulis deskripsi produk'.
- Harga:** A text input field with the value '0'.
- Kategori:** A dropdown menu with the placeholder '-- PILIH KATEGORI --'.
- Sub Kategori:** A dropdown menu with the placeholder '-- PILIH SUB KATEGORI --'.
- TAMBAH GAMBAR:** A button to upload an image.
- SIMPAN:** A purple button to save the product.

**Gambar 5.14. Potongan halaman tambah produk.**

```

public ResponseTemplate saveNewProduct(NewProductRequest
request) {
    BasicResult result = this
        .validationSaveNewProduct(request);

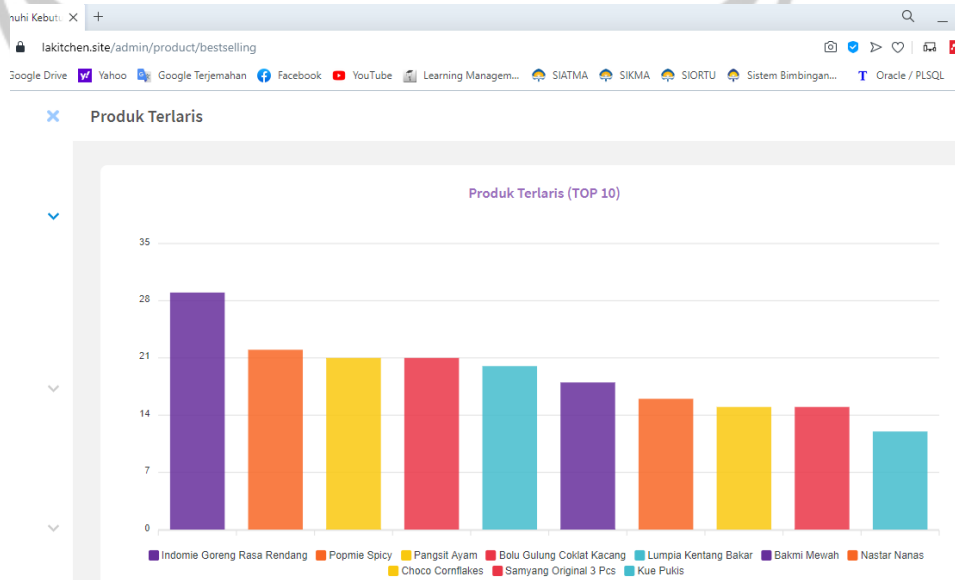
    if (result.getResult()) {
        Product product = new Product();
        Shop shop = shopRepository.findFirstById(1);
        ProductSubCategory subCategory =
            productSubCategoryRepository
                .findFirstById(request.getSubCategoryId());
        product.setSeen(0);
        product.setShop(shop);
        product.setName(request.getName());
        product.setPrice(request.getPrice());
        product.setProductSubCategory(subCategory);
        product.setDescription(request.getDescription());
        product.setCreatedAt(FUNC.getCurrentTimestamp());
        Product productSaved = productRepository.save(product);
    }
}

```

**Kode 5.14. Potongan kode tambah produk.**

#### 15) Halaman Laporan Produk Terlaris

Halaman ini hanya dapat diakses oleh *admin* untuk melihat produk yang paling banyak terjual. Seperti yang terlihat pada Gambar 5.15. dan Kode 5.15. bahwa sistem menampilkan 10 produk terlaris dari semua kategori dalam bentuk grafik batang. Selain itu, pengguna juga dapat melihat 5 produk terlaris berdasarkan kategori dalam bentuk tabel.



**Gambar 5.15. Potongan halaman laporan produk terlaris.**

```

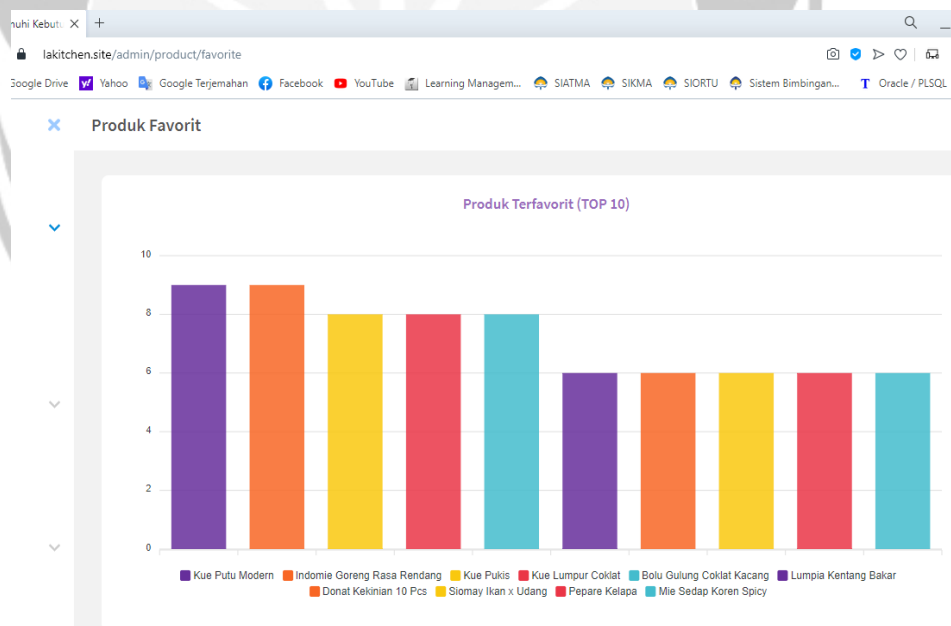
@Query(value =
"SELECT od.product_id as id, p.name, " +
"SUM(od.quantity) as sold " +
"FROM order_detail od JOIN products p " +
"ON (od.product_id = p.id) " +
"JOIN orders o ON (od.order_number = o.order_number) " +
"WHERE o.order_status_id = 5 " +
"GROUP BY od.product_id, p.name " +
"ORDER BY sold DESC LIMIT ?1", nativeQuery = true)
ArrayList<ProductTopSellingDTO>
findBestSellingProduct(Integer limit);

```

**Kode 5.15. Potongan kode laporan produk terlaris.**

## 16) Halaman Laporan Produk Terfavorit

Halaman ini hanya dapat diakses oleh *admin* untuk melihat produk yang paling disukai oleh pembeli. Seperti yang terlihat pada Gambar 5.16. dan Kode 5.16. bahwa sistem menampilkan 10 produk terfavorit dari semua kategori dalam bentuk grafik batang. Selain itu, pengguna juga dapat melihat 5 produk terfavorit berdasarkan kategori dalam bentuk tabel.



**Gambar 5.16. Potongan halaman laporan produk terfavorit.**

```

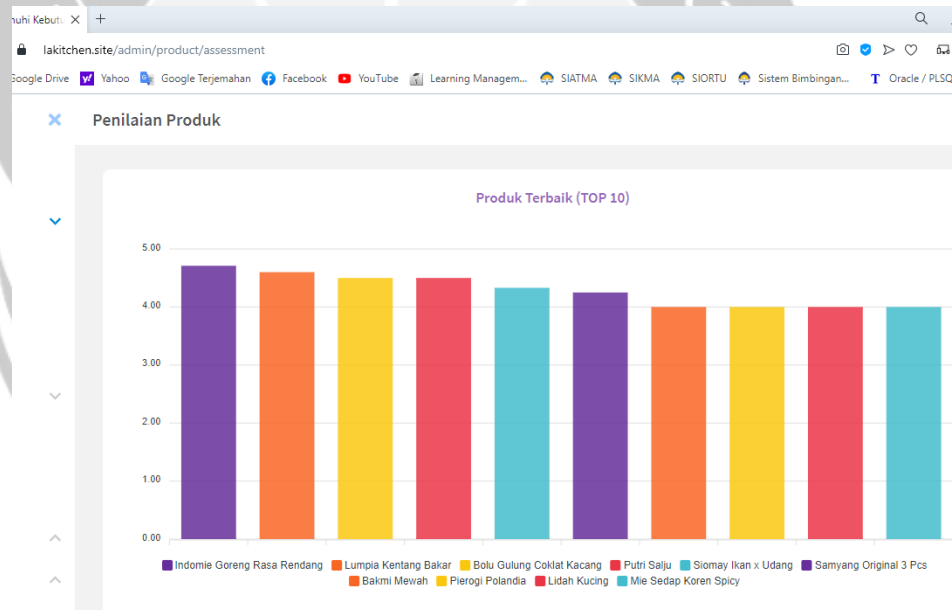
@Query(value =
"SELECT p.id, p.name, COUNT(w.product_id) as popularity " +
"FROM products p JOIN wishlists w " +
"ON (p.id = w.product_id " +
"GROUP BY w.product_id, p.name, p.id " +
"ORDER BY popularity DESC LIMIT ?1", nativeQuery = true)
ArrayList<ProductTopFavoriteDTO>
findBestFavoriteProduct(Integer limit);

```

**Kode 5.16. Potongan kode laporan produk terfavorit.**

17) Halaman Laporan Penilaian Produk

Halaman ini hanya dapat diakses oleh *admin* untuk melihat produk yang memiliki *rating* paling tinggi. Seperti yang terlihat pada Gambar 5.17. dan Kode 5.17. bahwa sistem menampilkan 10 produk terbaik dari semua kategori dalam bentuk grafik batang. Selain itu, pengguna juga dapat melihat 5 produk terbaik berdasarkan kategori dalam bentuk tabel.



**Gambar 5.17. Potongan halaman laporan penilaian produk.**

```

@Query(
value = "SELECT pa.product_id as id, p.name, " +
"ROUND(CAST(AVG(pa.rate) as numeric), 2) as rating " +
"FROM product_assessments pa JOIN products p " +
"ON (pa.product_id = p.id) " +
"GROUP BY pa.product_id, p.name ORDER BY rating DESC " +
"LIMIT ?1", nativeQuery = true)
ArrayList<ProductTopRatingDTO> findTopRatingProduct(Integer
limit);

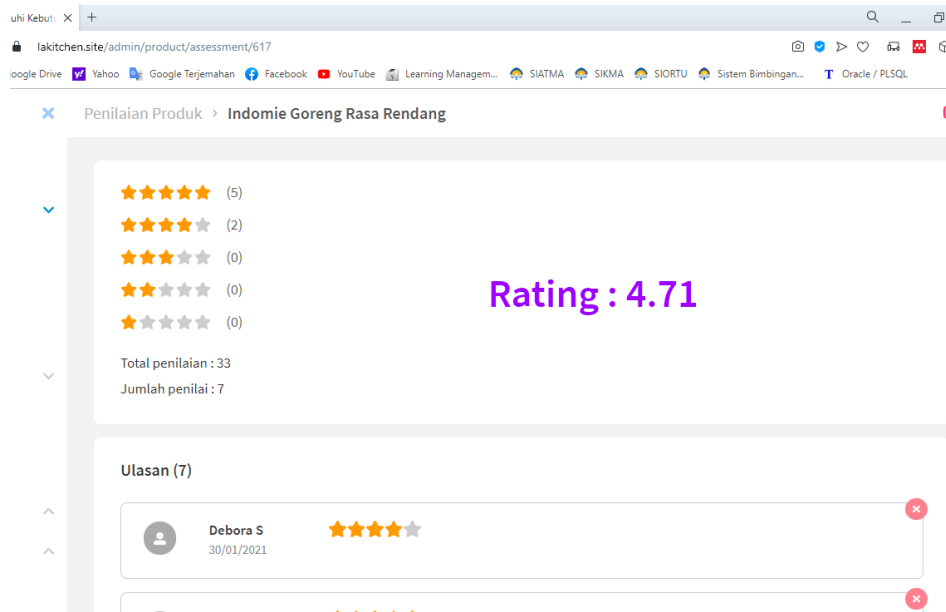
```

**Kode 5.17. Potongan kode laporan penilaian produk.**

18) Halaman Detail Penilaian Produk

Halaman ini digunakan oleh *admin* untuk melihat detail penilaian terhadap suatu produk tertentu. Seperti yang terlihat pada Gambar 5.18. bahwa sistem akan menampilkan rincian bintang yang diperoleh dari penilaian produk tersebut, yang kemudian dijumlah dan dibagi dengan total penilai untuk mendapatkan *rating* dari produk tersebut. Terlihat pada Kode 5.18. bahwa sistem *backend* memperoleh data - data yang diperlukan langsung dari *repository*. Kelas *repository* adalah kelas yang langsung berhubungan dengan basis data sistem, sehingga proses pengambilan, pengubahan, penghapusan, ataupun semua *query* lainnya terjadi di dalam kelas ini. Sistem *backend* akan mengambil satu per satu jumlah penilai yang ada di bintang penilaian, mulai dari bintang 5 - 1. Selain itu, sistem juga memberikan informasi terkait total komentar yang diperoleh. Pada halaman ini, pengguna juga dapat menghapus komentar yang dianggap tidak baik atau merugikan pihak LA' Kitchen. Komentar yang dihapus tidak akan ditampilkan pada halaman detail produk, akan tetapi penilaian berupa *rating* tetap masih dihitung dan ditampilkan.





**Gambar 5.18. Potongan halaman detail penilaian produk.**

```

public ResponseTemplate getByProductId(Integer productId) {
    Product product = productRepository
        .findFirstById(productId);
    if (product == null) {
        return new ResponseTemplate(404, "NOT FOUND", null,
            null, "Produk tidak ditemukan");
    }

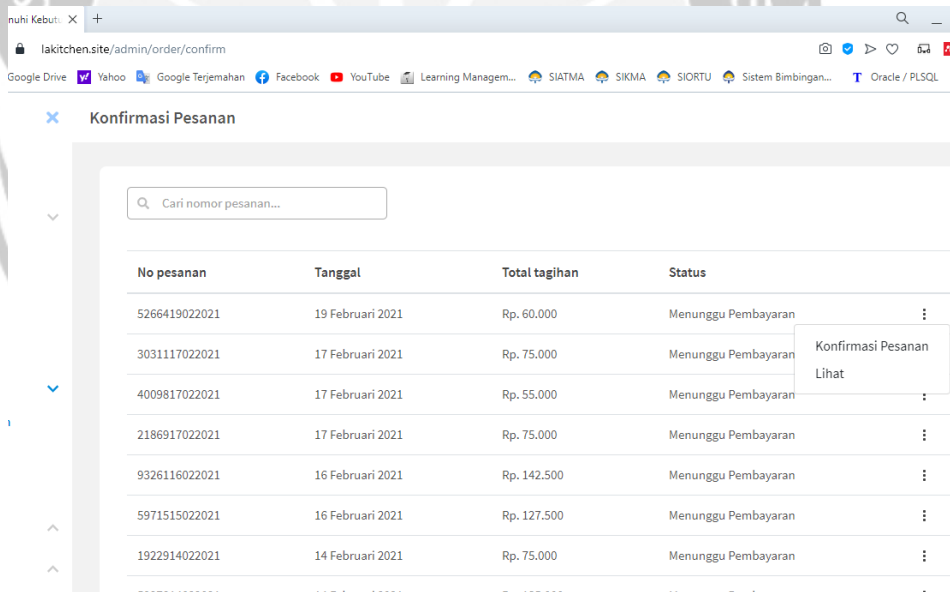
    DetailAssessmentDTO detailAssessmentDTO =
        assessmentRepository
            .findDetailAssessmentByProductId(productId);
    Integer rateFive = assessmentRepository
        .countSpecificRate(productId, 5);
    Integer rateFour = assessmentRepository
        .countSpecificRate(productId, 4);
    Integer rateThree = assessmentRepository
        .countSpecificRate(productId, 3);
    Integer rateTwo = assessmentRepository
        .countSpecificRate(productId, 2);
    Integer rateOne = assessmentRepository
        .countSpecificRate(productId, 1);
    GetByProductId response = new GetByProductId(
        product.getName(),
        assessmentMapper.mapToRateDTO(rateFive, rateFour,
            rateThree, rateTwo, rateOne),
        detailAssessmentDTO.getTotalAssessments(),
        detailAssessmentDTO.getEvaluators(),
        detailAssessmentDTO.getRating(),
        detailAssessmentDTO.getTotalComments());
}

```

**Kode 5.18. Potongan kode detail penilaian produk.**

## 19) Halaman Pengelolaan Transaksi

Halaman ini digunakan oleh *admin* untuk melakukan pengelolaan transaksi yang dilakukan oleh pembeli. Seperti yang terlihat pada Gambar 5.19. dan Kode 5.19. bahwa sistem menampilkan seluruh pesanan menggunakan *paging*, yaitu sepuluh data per *page*. Pengguna dapat melakukan konfirmasi pembayaran terhadap suatu pesanan jika pembeli telah melakukan pembayaran. Selain itu, pengguna juga dapat melakukan pencarian berdasarkan nomor pesanan. Pengguna juga dapat memperbarui status pesanan terhadap transaksi tertentu. Sistem juga menyediakan fitur untuk melihat pesanan berdasarkan status. Sistem juga menyimpan dan menampilkan riwayat transaksi yang dilakukan oleh penjual dan pembeli. Melalui halaman ini, pengguna juga dapat mengunjungi halaman detail transaksi untuk mengetahui seluruh data yang berkaitan dengan transaksi, seperti data pembeli, produk - produk yang dibeli, ataupun informasi transaksi itu sendiri.



No pesanan	Tanggal	Total tagihan	Status
5266419022021	19 Februari 2021	Rp. 60.000	Menunggu Pembayaran
3031117022021	17 Februari 2021	Rp. 75.000	Menunggu Pembayaran
4009817022021	17 Februari 2021	Rp. 55.000	Menunggu Pembayaran
2186917022021	17 Februari 2021	Rp. 75.000	Menunggu Pembayaran
9326116022021	16 Februari 2021	Rp. 142.500	Menunggu Pembayaran
5971515022021	16 Februari 2021	Rp. 127.500	Menunggu Pembayaran
1922914022021	14 Februari 2021	Rp. 75.000	Menunggu Pembayaran

**Gambar 5.19. Potongan halaman pengelolaan transaksi.**

```

public ResponseTemplate getByStatus(Integer page, Integer
statusId) {
    Pageable paging = PageRequest.of((page-1), 10,
Sort.by("createdAt").descending());
    Page<Order> orders = orderRepository
        .findByOrderStatus_Id(paging, statusId);
    ArrayList<OrderGeneralDTO> dto = this
        .helperMapToOrderGeneralDTO(orders);
    PageableDTO pageableDTO = FUNC
        .mapToPageableDTO(orders);
    return new ResponseTemplate(200, "OK",
        new GetByStatus(dto), pageableDTO, null);
}

```

### Kode 5.19. Potongan kode pengelolaan transaksi.

#### 20) Halaman Detail Transaksi

Halaman ini digunakan oleh *admin* untuk mengetahui detail informasi dari sebuah transaksi. Seperti yang terlihat pada Gambar 5.20. bahwa sistem menampilkan data - data yang berkaitan dengan transaksi. Pada Kode 5.20. selain mengambil data transaksi, sistem *backend* juga memberikan data pembeli yang dijadikan sebagai alamat pengiriman, produk - produk beserta jumlah belinya, catatan yang ditulis pada toko, ataupun informasi dari transaksi itu sendiri, yang diberikan kepada sistem *frontend*.

The screenshot shows a web browser window with the URL `lakitchen.site/admin/order/1948302032021`. The page content is as follows:

Data Customer		Data Pesanan	
Nama	Helga Karian	No pesanan	1948302032021
Email	usertest23@gmail.com	Tanggal	2 Maret 2021
No. HP	08127520231	Pembayaran	TRANSFER
Alamat	Jalan Maybara	Total tagihan	Rp. 225.000
Provinsi	Banten	Status	Siap Dikirim
Kota	Kab. Serang		

Pembelian Produk				
Produk	Harga	Jumlah beli	Sub total	Catatan
Kue Pukis	Rp. 35.000	3	Rp. 105.000	
Hakau Ayam x Udang	Rp. 52.500	2	Rp. 105.000	

Gambar 5.20. Potongan halaman detail transaksi.

```

public ResponseTemplate getById(String orderNumber) {
    BasicResult result = this
        .validationGetById(orderNumber);

    if (result.getResult()) {
        Order order = orderRepository
            .findFirstByOrderNumber(orderNumber);
        Payment payment = paymentRepository
            .findFirstByOrder_OrderNumber(orderNumber);
        User user = userRepository
            .findFirstById(order.getUser().getId());
        ArrayList<OrderDetail> orderDetails =
            orderDetailRepository
                .findByOrder_OrderNumber(orderNumber);
        OrderAdminDTO orderDTO = orderMapper
            .mapToOrderAdminDTO(order, payment);
        UserDTO userDTO = userMapper.mapToUserDTO(user);
        ArrayList<ProductOrderDTO> productOrderDTOS = this
            .helperMapToProductOrder(orderDetails);

        return new ResponseTemplate(200, "OK",
            new GetById(userDTO, orderDTO, productOrderDTOS),
            null, result.getError());
    }

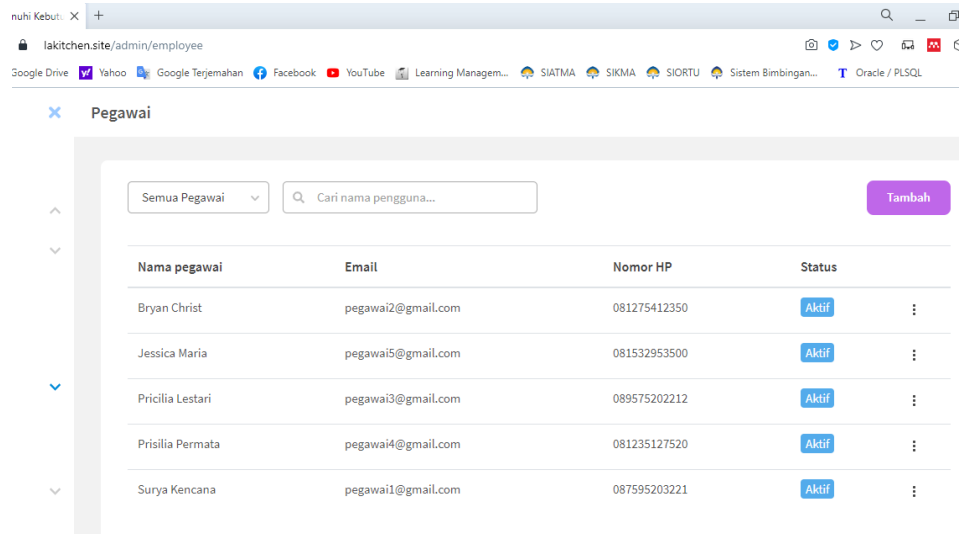
    return new ResponseTemplate(result.getCode(),
        result.getStatus(), null, null, result.getError());
}

```

**Kode 5.20. Potongan kode detail transaksi.**

21) Halaman Pengelolaan Pengguna

Halaman ini digunakan oleh *admin* untuk melakukan pengelolaan terhadap pengguna yang terdaftar pada sistem. Seperti yang terlihat pada Gambar 5.21. bahwa sistem menyediakan 2 fitur pengelolaan, yaitu pengelolaan *customer* dan pegawai. Sistem menampilkan seluruh data pengguna dalam bentuk tabel dan *paging*. Seperti yang terlihat pada Kode 5.21. sistem menggunakan *paging* yang terdiri dari 10 data per *page* dan diurutkan secara *ascending*, yaitu sesuai abjad dari a - z. Sistem juga menyediakan fitur *filter* untuk melihat data berdasarkan statusnya. Pengguna sistem memiliki 3 status, antara lain tidak aktif, aktif, dan diblokir. Sistem menyediakan fitur untuk melakukan blokir pada pengguna yang ingin dilarang masuk atau dianggap merugikan. Selain itu, sistem juga menyediakan fitur pencarian pengguna berdasarkan nama.



**Gambar 5.21. Potongan halaman pengelolaan pengguna.**

```

async getEmployee(page) {
  this.searchText = '';
  this.loader = true;

  let action;
  const params = { page };

  if (this.selectedStatus) {
    params.userStatusId = this.selectedStatus;
    action = this.getAllEmployeeByStatus;
  } else {
    action = this.getAllEmployee;
  }

  const { code, data, paging } = await this
    .$func.promiseAPI(action, params);

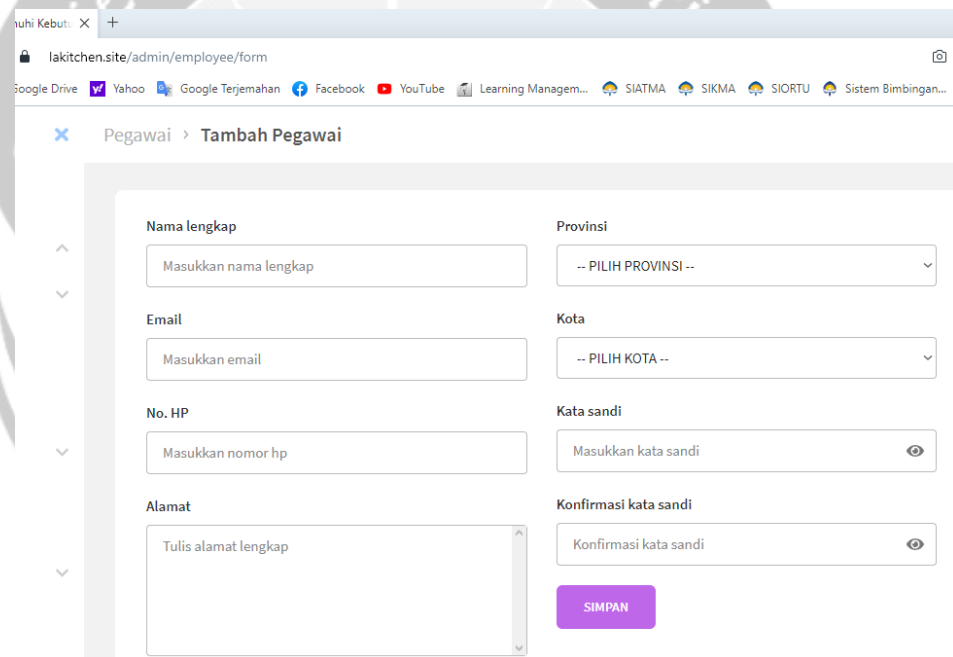
  this.loader = false;
  this.setData(code, data, paging, page);
}

```

**Kode 5.21. Potongan kode pengelolaan pengguna.**

## 22) Halaman Tambah Pegawai

Halaman ini berisi *form* yang dapat digunakan oleh *admin* untuk melakukan penambahan ataupun perubahan data pada akun pegawai. Seperti yang terlihat pada Gambar 5.22. sistem meminta pengguna untuk memasukkan beberapa data pribadi. Selain untuk kebutuhan *login*, hal ini juga prosedur umum yang perlu dilakukan dalam dunia pekerjaan, yaitu menyimpan data pegawai yang bekerja. Seperti yang terlihat pada Kode 5.22. bahwa sistem melakukan validasi terhadap data yang dimasukkan, kemudian menyimpannya dalam basis data sebagai *role customer service*. Sistem juga melakukan enkripsi terhadap kata sandi, dengan tujuan menghalangi orang mengetahui kata sandi tersebut ketika mengakses *database*.



The image shows a browser window displaying a web form titled "Tambah Pegawai" (Add Employee) on the URL "lakitchen.site/admin/employee/form". The form is organized into two columns and includes the following fields:

- Nama lengkap** (Full Name): A text input field with the placeholder "Masukkan nama lengkap".
- Provinsi** (Province): A dropdown menu with the placeholder "-- PILIH PROVINSI --".
- Email**: A text input field with the placeholder "Masukkan email".
- Kota** (City): A dropdown menu with the placeholder "-- PILIH KOTA --".
- No. HP** (Phone Number): A text input field with the placeholder "Masukkan nomor hp".
- Kata sandi** (Password): A text input field with the placeholder "Masukkan kata sandi" and an eye icon to toggle visibility.
- Alamat** (Address): A large text area with the placeholder "Tulis alamat lengkap".
- Konfirmasi kata sandi** (Confirm Password): A text input field with the placeholder "Konfirmasi kata sandi" and an eye icon to toggle visibility.

A purple button labeled "SIMPAN" (Save) is located at the bottom right of the form.

**Gambar 5.22. Potongan halaman tambah pegawai.**

```

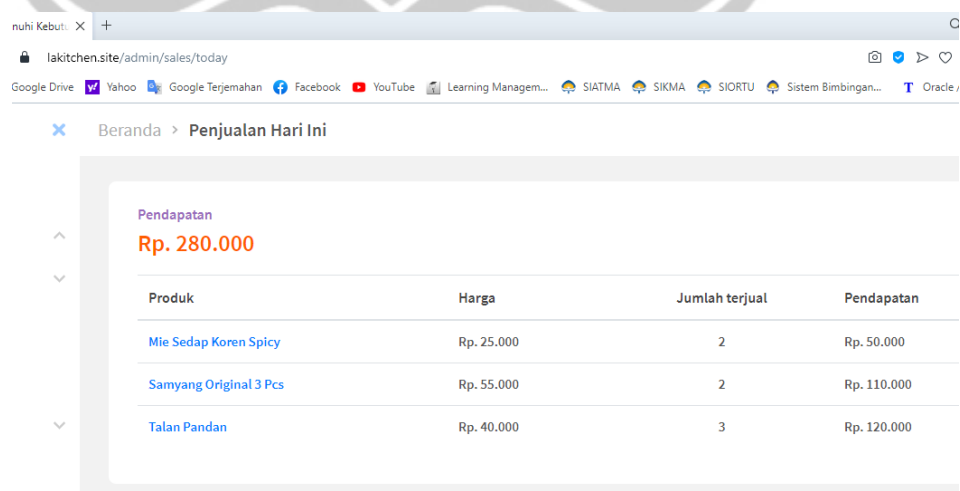
public ResponseTemplate createEmployee(NewUserRequest
request) {
    BasicResult result = this
        .validationCreateEmployee(request);
    if (result.getResult()) {
        User user = new User();
        user
            .setStatus(userStatusRepository.findFirstById(1));
        user.setUserRole(userRoleRepository.findFirstById(3));
        user.setName(request.getName());
        user.setEmail(request.getEmail());
        user.setPhoneNumber(request.getPhoneNumber());
        user.setAddress(request.getAddress());
        user.setProvince(request.getProvince());
        user.setCity(request.getCity());
        user
            .setPassword(this
                .BCryptEncoder(request.getPassword()));
        user.setCreatedAt(FUNC.getCurrentTimestamp());
        userRepository.save(user);
    }
}

```

**Kode 5.22. Potongan kode tambah pegawai.**

23) Halaman Laporan Penjualan Hari Ini

Halaman ini digunakan oleh *admin* untuk melihat laporan penjualan pada hari ini. Seperti yang terlihat pada Gambar 5.23. dan Kode 5.23. bahwa sistem menampilkan total pendapatan yang diperoleh hari ini beserta produk - produk yang terjual. Informasi produk yang ditampilkan seperti nama, harga, jumlah terjual, dan sub pendapatan dari produk tersebut.



Produk	Harga	Jumlah terjual	Pendapatan
Mie Sedap Koren Spicy	Rp. 25.000	2	Rp. 50.000
Samyang Original 3 Pcs	Rp. 55.000	2	Rp. 110.000
Talan Pandan	Rp. 40.000	3	Rp. 120.000

**Gambar 5.23. Potongan halaman laporan penjualan hari ini.**

```

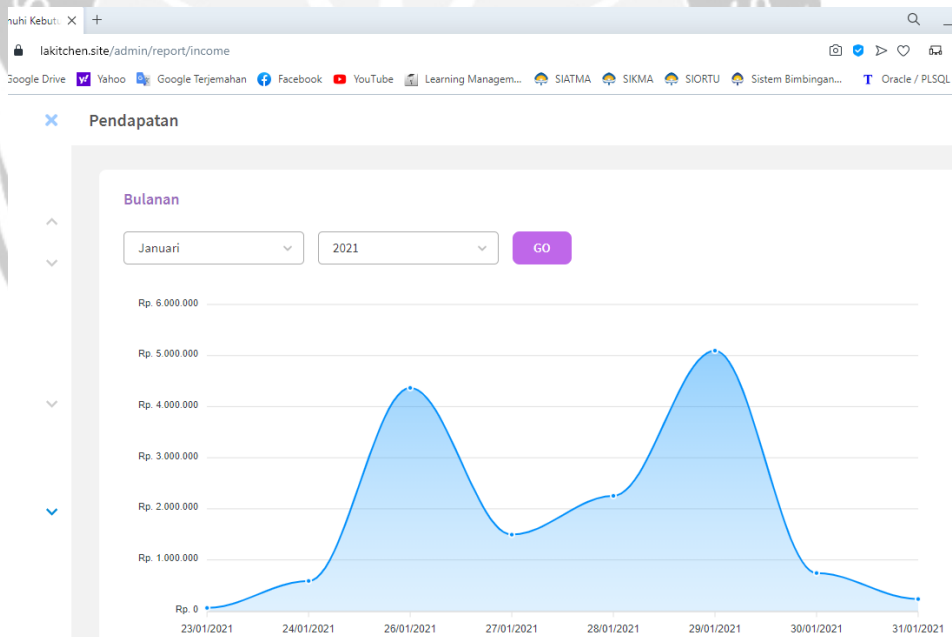
@Query(value =
"SELECT od.product_id as id, p.name, od.price, " +
"SUM(od.quantity) as sold FROM order_detail od " +
"JOIN products p ON (od.product_id = p.id) " +
"JOIN orders o ON (od.order_number = o.order_number) " +
"WHERE TO_DATE(CAST(o.paid_at as TEXT), 'YYYY-MM-DD') = " +
"CURRENT_DATE " +
"GROUP BY od.product_id, p.name, od.price " +
"ORDER BY p.name", nativeQuery = true)
ArrayList<SalesTodayDTO> findSalesToday();

```

**Kode 5.23. Potongan kode laporan penjualan hari ini.**

24) Halaman Laporan Pendapatan

Halaman ini digunakan oleh *admin* untuk melihat pendapatan yang diperoleh. Sistem menyediakan dua fitur laporan pendapatan, yaitu secara bulanan ataupun tahunan. Seperti yang terlihat pada Gambar 5.24. dan Kode 5.24. bahwa sistem akan meminta pengguna untuk memilih bulan atau tahun yang ingin dilihat. Setelah itu, sistem akan menampilkan pendapatan yang diperoleh dalam bentuk grafik garis.



**Gambar 5.24. Potongan halaman laporan pendapatan.**



```

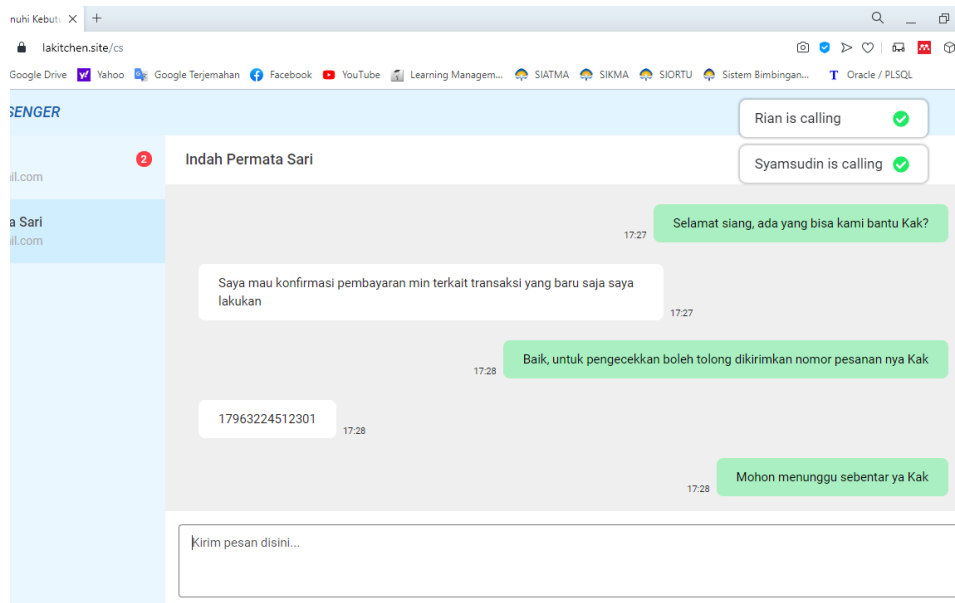
@Query(value =
"SELECT TO_DATE(CAST(o.paid_at as TEXT), 'YYYY-MM-DD') " +
"as createdAt, " +
"SUM(p.total) as income " +
"FROM payments p JOIN orders o " +
"ON (p.order_number = o.order_number) " +
"WHERE EXTRACT(YEAR FROM o.paid_at) = ?1 AND " +
"EXTRACT(MONTH FROM o.paid_at) = ?2 " +
"GROUP BY createdAt ORDER BY createdAt ASC",
nativeQuery = true)
ArrayList<Report2DTO> findMonthly(Integer year, Integer
month);

```

**Kode 5.24. Potongan kode laporan pendapatan.**

25) Halaman *Messenger*

Halaman ini digunakan oleh *customer service* untuk melakukan *live chat* yang dilakukan oleh *user*. Terlihat seperti Gambar 5.25. bahwa sistem akan menampilkan *calling* yang dilakukan oleh *user*, jika *customer service* menerimanya maka secara otomatis *user* dimasukkan ke dalam kontak dan dapat melakukan *live chat*. Fitur *live chat* disini tidak seperti fitur *chat* pada media sosial yang disimpan dan dapat terus dilihat di kemudian waktu. Fitur ini bersifat sementara, jika *user* atau *customer service* mengakhiri call maka *chat* akan segera hilang dari tampilan. Untuk melakukannya lagi, *user* perlu melakukan *calling* ulang kepada *customer service*. Terlihat pada Kode 5.25. bahwa fitur *live chat* merupakan salah satu fitur yang bertipe *real time app*. Maka dari itu jika fitur - fitur pada biasanya menggunakan HTTP, fitur *live chat* ini menggunakan WebSocket sebagai protokol komunikasi antara klien dengan server untuk menciptakan *real time app*. Sistem memiliki beberapa *channel* untuk melakukan *live chat*, contohnya sistem pada klien harus melakukan *subscribe* pada *channel* yang dibuat oleh sistem server untuk melakukan komunikasi. Obrolan yang telah dilakukan oleh *customer service* dan *user* akan disimpan pada basis data sistem.



**Gambar 5.25. Potongan halaman messenger.**

```

connection() {
  this.socket = new SockJS(api.sockURL);
  this.stompClient = Stomp.over(this.socket);
}
subscribeChatMessage(callId) {
  return this.stompClient.subscribe(`/message/${callId}`,
  (res) => {
    const incomingMessage = JSON.parse(res.body);
    if (this.selectedContact &&
      this.selectedContact.callId
      === incomingMessage.callId) {
      this.pushMessage(incomingMessage);
      this.readUnreadMessage();
    } else {
      const index = this
        .csContact
        .findIndex((val) =>
          val.callId === incomingMessage.callId);
      this.csContact[index].unreadMessages += 1;
      this.orderContact(incomingMessage.callId);
    }
  });
}
}

```

**Kode 5.25. Potongan kode live chat.**

## 5.2. Pengujian Fungsionalitas Perangkat Lunak

Pada subbagian ini berisikan tabel pengujian terhadap fungsionalitas sistem *e-commerce* yang dibangun pada LA' Kitchen. Pengujian ini dilakukan untuk memastikan bahwa sistem berjalan dengan baik. Pengujian tersebut dapat dilihat pada Tabel 5.1.

**Tabel 5.1. Hasil pengujian.**

ID	Deskripsi	Flow	Inputan	Expected Output	Actual Output	Kesimpulan
P01-01	<i>Login.</i>	<ol style="list-style-type: none"> <li>1) Masukkan <i>email</i>.</li> <li>2) Masukkan kata sandi.</li> <li>3) Tekan tombol "Masuk".</li> </ol>	<ul style="list-style-type: none"> <li>• <i>Email</i>: "albert@gmail.com".</li> <li>• Kata sandi: 123</li> </ul>	Sistem mengalihkan halaman ke <i>home</i> .	Sistem mengalihkan halaman ke <i>home</i> .	Handal.
P02-01	<i>Register.</i>	<ol style="list-style-type: none"> <li>1) Masukkan <i>email</i>.</li> <li>2) Masukkan nomor hp.</li> <li>3) Masukkan kata sandi.</li> <li>4) Masukkan konfirmasi kata sandi.</li> <li>5) Tekan tombol "Daftar".</li> </ol>	<ul style="list-style-type: none"> <li>• <i>Email</i>: "albert.kurniawan145@gmail.com".</li> <li>• Nomor hp: "08127528342"</li> <li>• Kata sandi: "123".</li> <li>• Konfirmasi kata sandi: "123".</li> </ul>	Sistem memberikan pesan bahwa pengguna harus melakukan verifikasi pada <i>email</i> .	Sistem memberikan pesan bahwa pengguna harus melakukan verifikasi pada <i>email</i> .	Handal.

P03-01	Pencarian produk.	<ol style="list-style-type: none"> <li>1) Masukkan nama produk yang ingin dicari.</li> <li>2) Tekan tombol “Cari”.</li> </ol>	<ul style="list-style-type: none"> <li>• Nama produk: “Mie”.</li> </ul>	Sistem menampilkan seluruh produk yang mengandung kata “Mie”.	Sistem menampilkan seluruh produk yang mengandung kata “Mie”.	Handal.
P04-01	Tampil berdasarkan kategori atau sub kategori.	<ol style="list-style-type: none"> <li>1) Klik “Kategori” pada <i>header</i>.</li> <li>2) Pilih dan klik kategori atau sub kategori yang diinginkan.</li> </ol>	-	Sistem menampilkan seluruh produk berdasarkan kategori atau sub kategori yang dipilih.	Sistem menampilkan seluruh produk berdasarkan kategori atau sub kategori yang dipilih.	Handal.
P04-02	Tampil berdasarkan harga terendah.	<ol style="list-style-type: none"> <li>1) Klik “Harga Terendah” pada <i>header</i>.</li> </ol>	-	Sistem menampilkan seluruh produk berdasarkan harga terendah.	Sistem menampilkan seluruh produk berdasarkan harga terendah.	Handal.
P04-03	Tampil berdasarkan harga tertinggi.	<ol style="list-style-type: none"> <li>1) Klik “Harga Tertinggi” pada <i>header</i>.</li> </ol>	-	Sistem menampilkan seluruh produk berdasarkan harga tertinggi.	Sistem menampilkan seluruh produk berdasarkan harga tertinggi.	Handal.

P05-01	Memperbarui profil.	<ol style="list-style-type: none"> <li>1) Masukkan nama lengkap.</li> <li>2) Masukkan alamat.</li> <li>3) Pilih provinsi.</li> <li>4) Pilih kota.</li> <li>5) Tekan tombol “Simpan”.</li> </ol>	<ul style="list-style-type: none"> <li>• Nama lengkap: “Albert K”.</li> <li>• Alamat pengiriman: “Jalan Sabara”.</li> <li>• Provinsi: “Sumatra Selatan”</li> <li>• Kota: “Palembang”.</li> </ul>	Sistem memperbarui data profil pengguna.	Sistem memperbarui data profil pengguna.	Handal.
P06-01	Menambahkan produk pada <i>wishlist</i> .	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin disimpan.</li> <li>2) Tekan ikon hati atau teks “Tambah ke Wishlist”.</li> </ol>	-	Sistem menampilkan pesan bahwa produk berhasil ditambahkan ke <i>wishlist</i> .	Sistem menampilkan pesan bahwa produk berhasil ditambahkan ke <i>wishlist</i> .	Handal.
P06-02	Menghapus produk dari <i>wishlist</i> .	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin dihapus.</li> <li>2) Tekan tombol “X” diatas kanan produk.</li> <li>3) Pilih “Ya”.</li> </ol>	-	Sistem menghapus dan memperbarui data produk di <i>wishlist</i> .	Sistem menghapus dan memperbarui data produk di <i>wishlist</i> .	Handal.

P07-01	Menambahkan produk ke <i>cart</i> .	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin ditambahkan.</li> <li>2) Tekan tombol “Tambah ke Keranjang”.</li> </ol>		Sistem menampilkan pesan berhasil serta ikon keranjang pada <i>header</i> bertambah.	Sistem menampilkan pesan berhasil serta ikon keranjang pada <i>header</i> bertambah.	Handal.
P07-02	Mengubah jumlah beli produk di <i>cart</i> .	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin diubah jumlah belinya.</li> <li>2) Klik tombol <i>plus</i> untuk tambah atau <i>minus</i> untuk mengurangi jumlah beli.</li> </ol>	-	Sistem memperbarui jumlah beli produk pada keranjang.	Sistem memperbarui jumlah beli produk pada keranjang.	Handal.
P07-03	Menambahkan catatan produk pada <i>cart</i> .	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin ditambahkan catatan.</li> <li>2) Klik teks “Tulis catatan”.</li> <li>3) Masukkan catatan untuk diketahui oleh toko.</li> <li>4) Tekan tombol “Simpan”.</li> </ol>	<ul style="list-style-type: none"> <li>• Catatan: “Minta topping yang banyak ya”.</li> </ul>	Sistem menyimpan dan menampilkan catatan produk pada keranjang.	Sistem menyimpan dan menampilkan catatan produk pada keranjang.	Handal.

P07-04	Menghapus produk dari <i>cart</i> .	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin dihapus.</li> <li>2) Tekan ikon tempat sampah.</li> <li>3) Pilih “Ya”.</li> </ol>	-	Sistem menampilkan pesan berhasil dan memperbarui data keranjang.	Sistem menampilkan pesan berhasil dan memperbarui data keranjang.	Handal.
P08-01	Buat pesanan.	<ol style="list-style-type: none"> <li>1) Pada keranjang, pilih produk yang ingin dibeli.</li> <li>2) Tekan tombol “Beli”.</li> <li>3) Tekan tombol “Bayar Sekarang”.</li> </ol>	-	Sistem menampilkan pesan melalui <i>popup</i> dan <i>email</i> terkait pembayaran.	Sistem menampilkan pesan melalui <i>popup</i> dan <i>email</i> terkait pembayaran.	Handal.
P08-02	Tampil daftar pesanan.	<ol style="list-style-type: none"> <li>1) Pilih “Saat Ini”, “Selesai”, atau “Dibatalkan”.</li> </ol>	-	Sistem menampilkan seluruh pesanan berdasarkan <i>state</i> yang dipilih pengguna.	Sistem menampilkan seluruh pesanan berdasarkan <i>state</i> yang dipilih pengguna.	Handal.
P08-03	Membatalkan pesanan dengan syarat status masih menunggu pembayaran.	<ol style="list-style-type: none"> <li>1) Pilih dan klik pesanan yang ingin dibatalkan.</li> <li>2) Tekan tombol “Batalkan pesanan”.</li> <li>3) Pilih “Ya”.</li> </ol>	-	Sistem menampilkan pesan bahwa pesanan berhasil dibatalkan.	Sistem menampilkan pesan bahwa pesanan berhasil dibatalkan.	Handal.

P09-01	Penilaian produk terhadap pesanan yang telah selesai.	<ol style="list-style-type: none"> <li>1) Tekan tombol “Nilai Produk” pada pesanan.</li> <li>2) Pilih produk yang ingin dinilai.</li> <li>3) Tekan tombol “Beri Nilai”.</li> <li>4) Masukkan jumlah bintang.</li> <li>5) Masukkan komentar.</li> </ol>	<ul style="list-style-type: none"> <li>• Jumlah bintang: “5”.</li> <li>• Komentar: “Enak sekali”.</li> </ul>	Sistem menampilkan pesan terimakasih karena telah memberikan penilaian pada produk LA’ Kitchen.	Sistem menampilkan pesan terimakasih karena telah memberikan penilaian pada produk LA’ Kitchen.	Handal.
P09-02	Tampil penilaian produk.	<ol style="list-style-type: none"> <li>1) Pilih dan masuk ke dalam detail produk.</li> <li>2) <i>Scroll</i> ke bawah dan pilih “Lihat Semua” pada bagian ulasan.</li> </ol>	-	Sistem menampilkan seluruh penilaian terhadap produk tersebut.	Sistem menampilkan seluruh penilaian terhadap produk tersebut.	Handal.
P10-01	Ganti kata sandi akun.	<ol style="list-style-type: none"> <li>1) Masukkan kata sandi lama.</li> <li>2) Masukkan kata sandi baru.</li> <li>3) Masukkan konfirmasi kata sandi baru.</li> <li>4) Tekan “Simpan”.</li> </ol>	<ul style="list-style-type: none"> <li>• Kata sandi lama: “123”.</li> <li>• Kata sandi baru: “456”.</li> <li>• Konfirmasi kata sandi baru: “456”.</li> </ul>	Sistem menampilkan pesan berhasil dan mengalihkan ke halaman <i>login</i> untuk melakukan autentikasi ulang.	Sistem menampilkan pesan berhasil dan mengalihkan ke halaman <i>login</i> untuk melakukan autentikasi ulang.	Handal.



P11-01	Tambah / ubah data produk.	<ol style="list-style-type: none"> <li>1) Masukkan nama produk.</li> <li>2) Masukkan harga.</li> <li>3) Pilih kategori.</li> <li>4) Pilih sub kategori.</li> <li>5) Masukkan deskripsi.</li> <li>6) Upload foto produk.</li> <li>7) Tekan tombol “Simpan”.</li> </ol>	<ul style="list-style-type: none"> <li>• Nama produk: “Mie Goreng”.</li> <li>• Harga: “25000”.</li> <li>• Kategori: “Mie &amp; Pasta”.</li> <li>• Sub kategori: “Mie Telur”.</li> <li>• Deskripsi: “Mie goreng telur homemade”.</li> <li>• Foto produk: “Mie.png”.</li> </ul>	Sistem menampilkan pesan bahwa produk berhasil disimpan dan mengalihkan halaman ke <i>product list</i> .	Sistem menampilkan pesan bahwa produk berhasil disimpan dan mengalihkan halaman ke <i>product list</i> .	Handal.
P11-02	Menonaktifkan / mengaktifkan produk.	<ol style="list-style-type: none"> <li>1) Pilih produk yang diinginkan.</li> <li>2) Tekan ikon pensil di paling kanan.</li> <li>3) Tekan tombol “Non Aktifkan” atau “Aktifkan”.</li> <li>4) Pilih “Ya”.</li> </ol>	-	Sistem menampilkan pesan berhasil dan mengalihkan halaman ke <i>product list</i> .	Sistem menampilkan pesan berhasil dan mengalihkan halaman ke <i>product list</i> .	Handal.

P11-03	Tampil seluruh produk.	1) Pilih tab “Produk”, kemudian pilih “Jual Produk”.	-	Sistem menampilkan seluruh produk yang terdaftar menggunakan <i>paging</i> .	Sistem menampilkan seluruh produk yang terdaftar menggunakan <i>paging</i> .	Handal.
P11-04	Tampil seluruh produk berdasarkan kategori.	1) Pilih tab “Produk”, kemudian pilih “Jual Produk”. 2) Pilih kategori yang diinginkan.	<ul style="list-style-type: none"> <li>• Kategori: “Kue”.</li> </ul>	Sistem menampilkan seluruh produk berdasarkan kategori kue menggunakan <i>paging</i> .	Sistem menampilkan seluruh produk berdasarkan kategori kue menggunakan <i>paging</i> .	Handal.
P11-05	Pencarian produk berdasarkan nama.	1) Pilih tab “Produk”, kemudian pilih “Jual Produk”. 2) Masukkan nama pada <i>search</i> .	<ul style="list-style-type: none"> <li>• Nama: “bolu”.</li> </ul>	Sistem menampilkan seluruh produk yang mengandung kata “bolu”.	Sistem menampilkan seluruh produk yang mengandung kata “bolu”.	Handal.
P12-01	Tampil seluruh pesanan yang perlu dikonfirmasi.	1) Pilih tab “Transaksi”, kemudian pilih “Konfirmasi Pesanan”.	-	Sistem menampilkan seluruh pesanan yang perlu dikonfirmasi penjual.	Sistem menampilkan seluruh pesanan yang perlu dikonfirmasi penjual.	Handal.

P12-02	Melakukan konfirmasi pesanan pembeli.	<ol style="list-style-type: none"> <li>1) Pilih tab “Transaksi”, kemudian pilih “Konfirmasi Pesanan”.</li> <li>2) Pilih pesanan yang ingin dikonfirmasi.</li> <li>3) Klik ikon bar yang ada di kanan.</li> <li>4) Pilih “Konfirmasi Pesanan”.</li> <li>5) Pilih “Ya”.</li> </ol>	-	Sistem menampilkan pesan bahwa pesanan berhasil dikonfirmasi dan menjadi pesanan yang aktif.	Sistem menampilkan pesan bahwa pesanan berhasil dikonfirmasi dan menjadi pesanan yang aktif.	Handal.
P12-03	Tampil seluruh pesanan yang sedang aktif.	<ol style="list-style-type: none"> <li>1) Pilih tab “Transaksi”, kemudian pilih “Pesanan Aktif”.</li> </ol>	-	Sistem menampilkan seluruh pesanan yang sedang aktif.	Sistem menampilkan seluruh pesanan yang sedang aktif.	Handal.
P12-04	Memperbarui status pesanan aktif.	<ol style="list-style-type: none"> <li>1) Pilih tab “Transaksi”, kemudian pilih “Pesanan Aktif”.</li> <li>2) Pilih pesanan yang ingin diperbarui.</li> <li>3) Klik ikon bar yang ada di kanan.</li> </ol>	<ul style="list-style-type: none"> <li>• Status pesanan: “Siap Dikirim”.</li> </ul>	Sistem menampilkan pesan bahwa status pesanan berhasil diperbarui.	Sistem menampilkan pesan bahwa status pesanan berhasil diperbarui.	Handal.

		<ol style="list-style-type: none"> <li>4) Pilih “Perbarui Status”.</li> <li>5) Pilih dan ubah status pesanan yang diinginkan.</li> <li>6) Pilih “Ok”.</li> </ol>				
P12-05	Tampil seluruh pesanan aktif berdasarkan status.	<ol style="list-style-type: none"> <li>1) Pilih tab “Transaksi”, kemudian pilih “Pesanan Aktif”.</li> <li>2) Pilih status pesanan.</li> </ol>	<ul style="list-style-type: none"> <li>• Status pesanan: “Dalam Pengiriman”.</li> </ul>	Sistem menampilkan seluruh pesanan aktif yang sedang dalam pengiriman.	Sistem menampilkan seluruh pesanan aktif yang sedang dalam pengiriman.	Handal.
P12-06	Tampil seluruh riwayat pesanan.	<ol style="list-style-type: none"> <li>1) Pilih tab “Transaksi”, kemudian pilih “Riwayat Pesanan”.</li> </ol>	-	Sistem menampilkan seluruh riwayat pesanan.	Sistem menampilkan seluruh riwayat pesanan.	Handal.
P12-07	Tampil detail pesanan.	<ol style="list-style-type: none"> <li>1) Pilih pesanan yang ingin dilihat.</li> <li>2) Klik ikon bar yang ada di kanan.</li> <li>3) Pilih “Lihat”.</li> </ol>	-	Sistem menampilkan detail pesanan yang meliputi data diri pembeli, informasi pesanan, dan produk yang dibeli.	Sistem menampilkan detail pesanan yang meliputi data diri pembeli, informasi pesanan, dan produk yang dibeli.	Handal.

P12-08	Pencarian pesanan berdasarkan nomor.	1) Masukkan nomor pesanan yang ingin dicari pada <i>search</i> .	<ul style="list-style-type: none"> <li>Nomor pesanan: "1602".</li> </ul>	Sistem menampilkan seluruh pesanan yang mengandung "1602".	Sistem menampilkan seluruh pesanan yang mengandung "1602".	Handal.
P13-01	Tampil seluruh pengguna sistem.	1) Pilih tab "Pengguna", kemudian pilih "Customer" atau "Pegawai".	-	Sistem menampilkan seluruh pengguna.	Sistem menampilkan seluruh pengguna.	Handal.
P13-02	Tampil seluruh pengguna berdasarkan status.	1) Pilih tab "Pengguna", kemudian pilih "Customer" atau "Pegawai". 2) Pilih status yang berada di atas tabel.	<ul style="list-style-type: none"> <li>Status: "Diblokir".</li> </ul>	Sistem menampilkan seluruh pengguna yang diblokir.	Sistem menampilkan seluruh pengguna yang diblokir.	Handal.
P13-03	Pencarian pengguna berdasarkan nama.	1) Masukkan nama pengguna pada <i>search</i> .	<ul style="list-style-type: none"> <li>Nama: "Rudi".</li> </ul>	Sistem menampilkan seluruh pengguna yang mengandung nama "Rudi".	Sistem menampilkan seluruh pengguna yang mengandung nama "Rudi".	Handal.

P13-04	Tampil detail data pengguna.	<ol style="list-style-type: none"> <li>1) Pilih pengguna yang ingin dilihat.</li> <li>2) Klik ikon bar yang ada di kanan.</li> <li>3) Pilih “Lihat”.</li> </ol>	-	Sistem menampilkan modal yang berisi detail data diri pengguna.	Sistem menampilkan modal yang berisi detail data diri pengguna.	Handal.
P13-05	Blokir akun pengguna.	<ol style="list-style-type: none"> <li>1) Pilih pengguna yang ingin diblokir.</li> <li>2) Klik ikon bar yang ada di kanan.</li> <li>3) Pilih “Blokir Akun”.</li> <li>4) Pilih “Ya”.</li> </ol>	-	Sistem menampilkan pesan bahwa akun tersebut berhasil diblokir.	Sistem menampilkan pesan bahwa akun tersebut berhasil diblokir.	Handal.
P13-06	Tambah / ubah data pegawai.	<ol style="list-style-type: none"> <li>1) Masukkan nama lengkap.</li> <li>2) Masukkan <i>email</i>.</li> <li>3) Masukkan nomor hp.</li> <li>4) Masukkan alamat.</li> <li>5) Pilih provinsi.</li> <li>6) Pilih kota.</li> <li>7) Masukkan kata sandi.</li> </ol>	<ul style="list-style-type: none"> <li>• Nama lengkap: “Kevin”.</li> <li>• <i>Email</i>: “kevins@gmail.com”.</li> <li>• Nomor hp: “08161752121”</li> <li>• Alamat: “Jalan Ridak No 4A”.</li> <li>• Provinsi: “DKI</li> </ul>	Sistem menampilkan pesan berhasil dan mengalihkan ke halaman tampil pegawai.	Sistem menampilkan pesan berhasil dan mengalihkan ke halaman tampil pegawai.	Handal.

		8) Masukkan konfirmasi kata sandi. 9) Tekan tombol “Simpan”.	Jakarta”. • Kota: “Jakarta Pusat”. • Kata sandi: “123”. • Konfirmasi kata sandi: “123”.			
P13-07	Reset password akun pegawai.	1) Pilih akun pegawai yang ingin di reset. 2) Klik ikon bar yang ada di kanan. 3) Pilih “Reset Password”. 4) Pilih “Ya”.	-	Sistem menampilkan pesan bahwa <i>password</i> berhasil di <i>reset</i> .	Sistem menampilkan pesan bahwa <i>password</i> berhasil di <i>reset</i> .	Handal.
P14-01	Tampil laporan produk terlaris.	1) Pilih tab “Produk”, kemudian pilih “Produk Terlaris”.	-	Sistem menampilkan grafik batang yang berisi 10 produk terlaris dari semua kategori.	Sistem menampilkan grafik batang yang berisi 10 produk terlaris dari semua kategori.	Handal.

P14-02	Tampil laporan produk terfavorit.	1) Pilih tab “Produk”, kemudian pilih “Produk Favorit”.	-	Sistem menampilkan grafik batang yang berisi 10 produk terfavorit dari semua kategori.	Sistem menampilkan grafik batang yang berisi 10 produk terfavorit dari semua kategori.	Handal.
P14-03	Tampil laporan produk terbaik.	1) Pilih tab “Produk”, kemudian pilih “Penilaian Produk”.	-	Sistem menampilkan grafik batang yang berisi 10 produk yang memiliki <i>rating</i> paling tinggi dari semua kategori.	Sistem menampilkan grafik batang yang berisi 10 produk yang memiliki <i>rating</i> paling tinggi dari semua kategori.	Handal.
P14-04	Tampil laporan penjualan hari ini.	1) Pilih tab “Beranda”, kemudian klik “Lihat” pada bagian penjualan hari ini.	-	Sistem menampilkan total pendapatan dan produk - produk yang terjual hari ini dalam bentuk tabel.	Sistem menampilkan total pendapatan dan produk - produk yang terjual hari ini dalam bentuk tabel.	Handal.
P14-05	Tampil laporan pendapatan satu minggu terakhir.	1) Lihat bagian pendapatan satu minggu terakhir di tab “Beranda”.	-	Sistem menampilkan pendapatan dalam bentuk grafik garis.	Sistem menampilkan pendapatan dalam bentuk grafik garis.	Handal.



P14-06	Tampil laporan pendapatan bulanan.	<ol style="list-style-type: none"> <li>1) Pilih tab “Laporan”, kemudian pilih “Pendapatan”.</li> <li>2) Pilih bulan dan tahun yang ingin dilihat.</li> </ol>	<ul style="list-style-type: none"> <li>• Bulan: “Februari”.</li> <li>• Tahun: “2021”.</li> </ul>	Sistem menampilkan total pendapatan per hari di bulan Februari 2021 dalam bentuk grafik garis.	Sistem menampilkan total pendapatan per hari di bulan Februari 2021 dalam bentuk grafik garis..	Handal.
P14-07	Tampil laporan pendapatan tahunan.	<ol style="list-style-type: none"> <li>1) Pilih tab “Laporan”, kemudian pilih “Pendapatan”.</li> <li>2) Pilih tahun yang ingin dilihat.</li> </ol>	<ul style="list-style-type: none"> <li>• Tahun: “2021”.</li> </ul>	Sistem menampilkan pendapatan per bulan di tahun 2021 dalam bentuk grafik garis.	Sistem menampilkan pendapatan per bulan di tahun 2021 dalam bentuk grafik garis.	Handal.
P15-01	Tampil detail penilaian produk.	<ol style="list-style-type: none"> <li>1) Pilih produk yang ingin dilihat.</li> <li>2) Tekan ikon bintang yang ada di ujung kanan.</li> </ol>	-	Sistem menampilkan detail penilaian produk seperti jumlah bintang dan ulasan.	Sistem menampilkan detail penilaian produk seperti jumlah bintang dan ulasan.	Handal.

P15-02	Hapus komentar dari penilaian produk.	<ol style="list-style-type: none"> <li>1) Masuk ke dalam detail penilaian produk.</li> <li>2) Pilih ulasan yang ingin dihapus.</li> <li>3) Tekan tombol “X” yang ada di atas kanan komentar.</li> <li>4) Pilih “Ya”.</li> </ol>	-	Sistem menghapus komentar dan tetap menampilkan penilaian <i>user</i> dalam bentuk <i>rating</i> .	Sistem menghapus komentar dan tetap menampilkan penilaian <i>user</i> dalam bentuk <i>rating</i> .	Handal.
P16-01	Melakukan panggilan ke <i>customer service</i>	<ol style="list-style-type: none"> <li>1) Buka “Chat” yang ada di <i>home</i>.</li> <li>2) Tekan “Mulai Percakapan”.</li> </ol>	-	Sistem menyimpan dan menampilkan panggilan pada halaman <i>customer service</i> .	Sistem menyimpan dan menampilkan panggilan pada halaman <i>customer service</i> .	Handal.
P16-02	Melakukan <i>live chat</i>	<ol style="list-style-type: none"> <li>1) Masukkan pesan yang ingin dikirim.</li> <li>2) Tekan <i>enter</i>.</li> </ol>	<ul style="list-style-type: none"> <li>• Pesan: “Selamat siang”.</li> </ul>	Sistem menyimpan dan menampilkan <i>live chat</i> kepada <i>user</i> dan CS.	Sistem menyimpan dan menampilkan <i>live chat</i> kepada <i>user</i> dan CS.	Handal.

### 5.3. Hasil Pengujian Terhadap Pengguna

Pada subbagian ini berisikan hasil dari penyebaran kuesioner yang dilakukan secara *online* kepada pengguna aplikasi. Pengujian dan kuesioner dilakukan terhadap pengguna aplikasi dari sisi pembeli dan penjual, sehingga terdapat dua kuesioner yang berbeda dari sudut pandang pembeli dan penjual. Hasil dari kuesioner ini adalah untuk membuktikan tercapainya atau tidak tujuan dari pembuatan sistem informasi *e-commerce* berbasis *website* pada LA' Kitchen. Pada pengujian ini, penulis menggunakan Skala Likert sebagai metode analisis yang digunakan untuk mengukur tanggapan positif dan negatif terhadap suatu pernyataan. Kuesioner yang disebarkan kepada pembeli terdiri dari sembilan pernyataan mengenai kepuasan pengguna dan satu isian singkat terkait saran bagi sistem LA' Kitchen. Sedangkan pada kuesioner penjual, terdiri dari delapan pernyataan mengenai kepuasan pengguna. Tipe pernyataan pada kuesioner menggunakan skala yang memiliki nilai satu sampai lima, dimulai dari satu yang merupakan sangat tidak setuju, hingga yang kelima yang merupakan sangat setuju. Rumus perhitungan Skala Likert ini dapat dilihat pada Persamaan (1) sampai dengan Persamaan (3).

#### Rumus Skala Likert :

$$\text{Skor Minimum} = \text{jumlah responden} \times \text{skala terendah} \quad (1)$$

$$\text{Skor Maksimum} = \text{jumlah responden} \times \text{skala tertinggi} \quad (2)$$

$$\text{Indeks (\%)} = (\text{total skor} / \text{skor maksimum}) \times 100 \quad (3)$$

#### Interval Penilaian :

Indeks 0% - 19.99% : Sangat tidak setuju

Indeks 20% - 39.99% : Tidak setuju

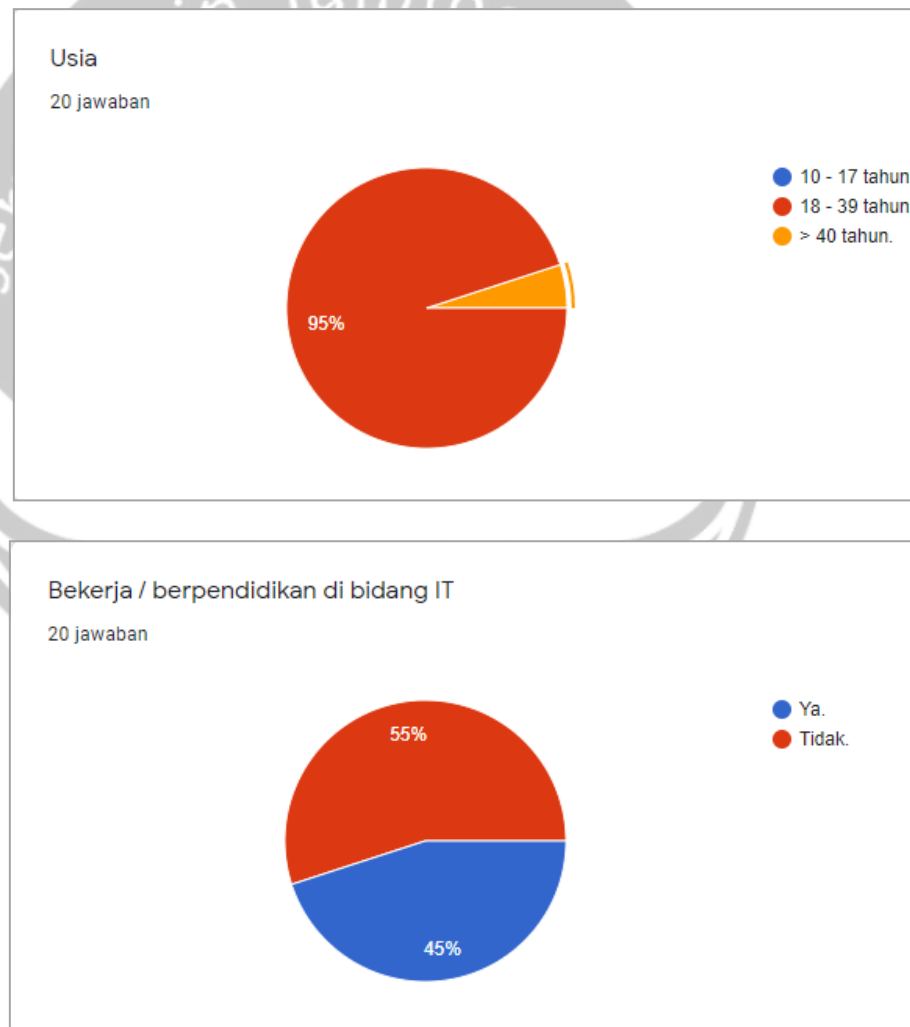
Indeks 40% - 59.99% : Ragu - ragu

Indeks 60% - 79.99% : Setuju

Indeks 80% - 100% : Sangat setuju

### 5.3.1. Kuesioner Pembeli

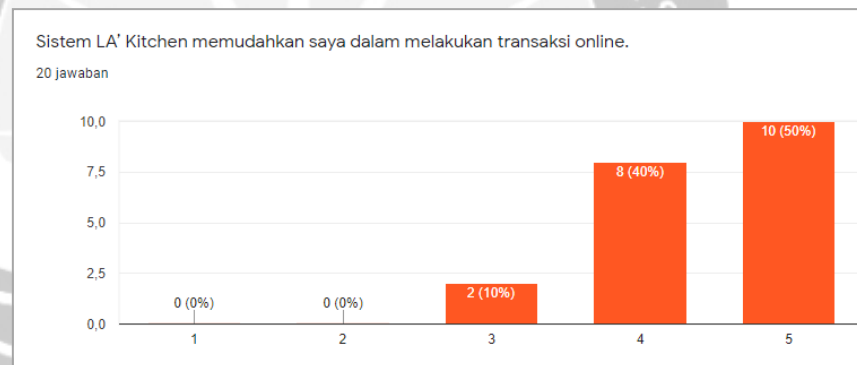
Dari penyebaran kuesioner, penulis memperoleh 20 responden yang telah mencoba dan menggunakan sistem LA' Kitchen. Seperti yang terlihat pada Gambar 5.26. bahwa 95% dari responden adalah berusia 18 - 39 tahun dan hanya 5% yang berusia diatas 40 tahun. Berdasarkan kuesioner tersebut, penulis memperoleh data bahwa 55% responden tidak memiliki latar belakang di bidang teknologi, sedangkan 45% sisanya memiliki latar belakang di bidang teknologi.



**Gambar 5.26. Usia dan latar belakang teknologi.**

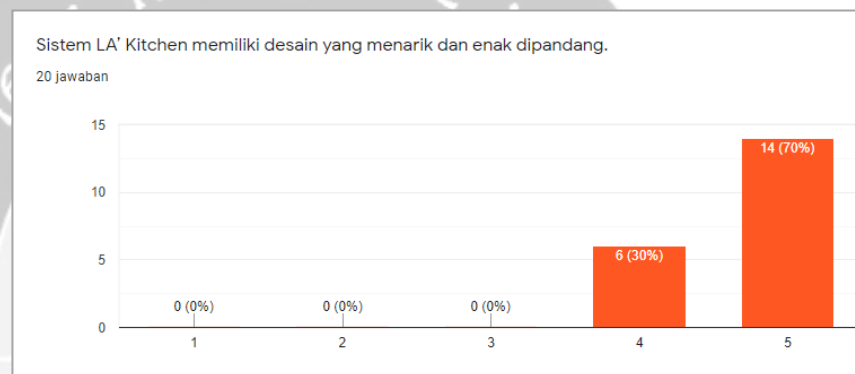
Berikut ini merupakan hasil skala terhadap pernyataan kuesioner dari responden yang telah mencoba dan menggunakan sistem.

- 1) Sistem LA' Kitchen memudahkan saya dalam melakukan transaksi *online*. Seperti yang terlihat pada Gambar 5.27. penulis menerima 10 responden yang menjawab sangat setuju, 8 responden menjawab setuju, dan 2 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 10) + (4 \times 8) + (2 \times 3)}{100} \times 100\% = 88\%$ . Berdasarkan interval indeks penilaian dari pernyataan pertama, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memudahkan pengguna dalam melakukan transaksi *online*.



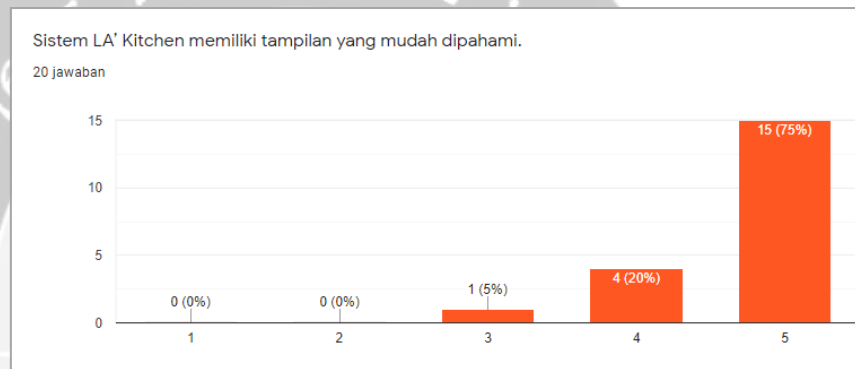
**Gambar 5.27. Kuesioner pembeli pernyataan pertama.**

2) Sistem LA' Kitchen memiliki desain yang menarik dan enak dipandang. Seperti yang terlihat pada Gambar 5.28. penulis menerima 14 responden yang menjawab sangat setuju, dan 6 responden menjawab setuju. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 14) + (4 \times 6)}{100} \times 100\% = 94\%$ . Berdasarkan interval indeks penilaian dari pernyataan kedua, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memiliki desain yang menarik dan enak dipandang.



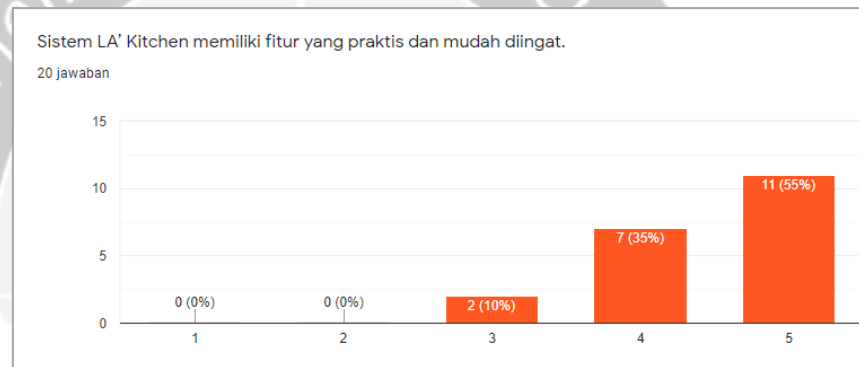
**Gambar 5.28. Kuesioner pembeli pernyataan kedua.**

3) Sistem LA' Kitchen memiliki tampilan yang mudah dipahami. Seperti yang terlihat pada Gambar 5.29. penulis menerima 15 responden yang menjawab sangat setuju, 4 responden menjawab setuju, dan 1 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 15) + (4 \times 4) + (3 \times 1)}{100} \times 100\% = 94\%$ . Berdasarkan interval indeks penilaian dari pernyataan ketiga, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memiliki tampilan yang mudah dipahami.



**Gambar 5.29. Kuesioner pembeli pernyataan ketiga.**

4) Sistem LA' Kitchen memiliki fitur yang praktis dan mudah diingat. Seperti yang terlihat pada Gambar 5.30. penulis menerima 11 responden yang menjawab sangat setuju, 7 responden menjawab setuju, dan 2 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 11) + (4 \times 7) + (3 \times 2)}{100} \times 100\% = 89\%$ . Berdasarkan interval indeks penilaian dari pernyataan keempat, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memiliki fitur yang praktis dan mudah diingat.



**Gambar 5.30. Kuesioner pembeli pernyataan keempat.**

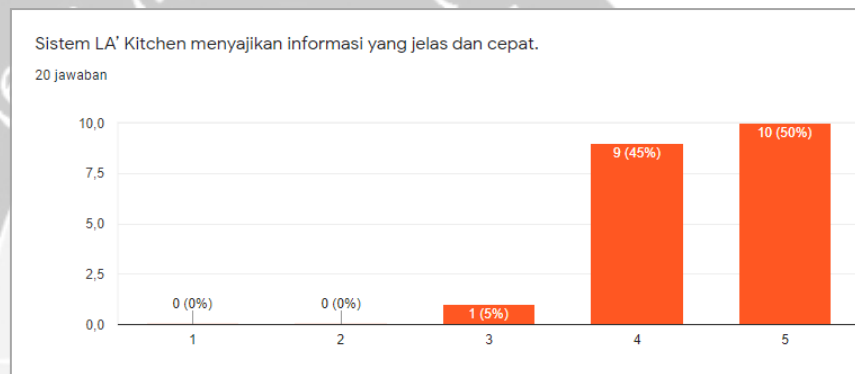


5) Sistem LA' Kitchen memiliki perintah dan peringatan yang jelas kepada pengguna. Seperti yang terlihat pada Gambar 5.31. penulis menerima 11 responden yang menjawab sangat setuju, dan 9 responden menjawab setuju. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 11) + (4 \times 9)}{100} \times 100\% = 91\%$ . Berdasarkan interval indeks penilaian dari pernyataan kelima, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memiliki perintah dan peringatan yang jelas kepada pengguna.



**Gambar 5.31. Kuesioner pembeli pernyataan kelima.**

6) Sistem LA' Kitchen menyajikan informasi yang jelas dan cepat. Seperti yang terlihat pada Gambar 5.32. penulis menerima 10 responden yang menjawab sangat setuju, 9 responden menjawab setuju, dan 1 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 10) + (4 \times 9) + (3 \times 1)}{100} \times 100\% = 89\%$ . Berdasarkan interval indeks penilaian dari pernyataan keenam, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen menyajikan informasi yang jelas dan cepat.



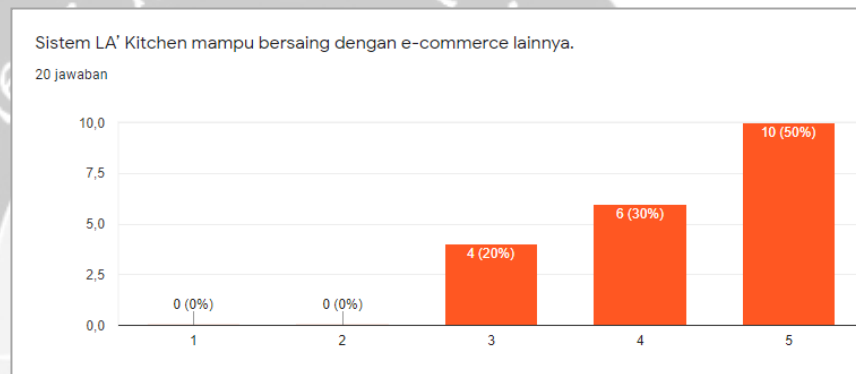
**Gambar 5.32. Kuesioner pembeli pernyataan keenam.**

7) Sistem LA' Kitchen memiliki daya tarik yang besar bagi pengguna. Seperti yang terlihat pada Gambar 5.33. penulis menerima 12 responden yang menjawab sangat setuju, 6 responden menjawab setuju, dan 2 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 12) + (4 \times 6) + (3 \times 2)}{100} \times 100\% = 90\%$ . Berdasarkan interval indeks penilaian dari pernyataan ketujuh, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memiliki daya tarik yang besar bagi pengguna.



**Gambar 5.33. Kuesioner pembeli pernyataan ketujuh.**

8) Sistem LA' Kitchen mampu bersaing dengan *e-commerce* lainnya. Seperti yang terlihat pada Gambar 5.34. penulis menerima 10 responden yang menjawab sangat setuju, 6 responden menjawab setuju, dan 4 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 10) + (4 \times 6) + (3 \times 4)}{100} \times 100\% = 86\%$ . Berdasarkan interval indeks penilaian dari pernyataan kedelapan, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen mampu bersaing dengan *e-commerce* lainnya.



**Gambar 5.34. Kuesioner pembeli pernyataan kedelapan.**

9) Sistem LA' Kitchen memberikan manfaat yang besar bagi saya. Seperti yang terlihat pada Gambar 5.35. penulis menerima 8 responden yang menjawab sangat setuju, 8 responden menjawab setuju, dan 4 responden menjawab ragu - ragu. Maka dari itu, berdasarkan perhitungan indeks Skala Likert diperoleh sebesar  $\frac{(5 \times 8) + (4 \times 8) + (3 \times 4)}{100} \times 100\% = 84\%$ . Berdasarkan interval indeks penilaian dari pernyataan kesembilan, penulis memperoleh hasil sangat setuju terhadap pernyataan bahwa sistem LA' Kitchen memberikan manfaat yang besar bagi pengguna.



**Gambar 5.35. Kuesioner pembeli pernyataan kesembilan.**

10) Saran untuk sistem *e-commerce* LA' Kitchen.

Berikut ini merupakan pesan dan saran dari responden yang telah mencoba dan menggunakan sistem.

1. Website sangat menarik dan interaktif.
2. Maju terus, semoga berkembang dengan pesat, bisa memberkati banyak orang.
3. Keren, sudah menyesuaikan layout HP juga walaupun berbasis web.
4. Mungkin untuk buttonnya bisa dipastiin lagi bisa di klik dan direct sesuai tujuannya, soalnya pas coba masih baada yang ga bisa di klik dan ada button yang buat bingung kayak burger menu disebelah kategori.
5. Kadang kadang tombol bantuannya tidak bisa di klik.
6. Pengecekan notifikasi pembayaran dan proses melakukan pembayaran agak ribet. Saran dibuat menu untuk melakukan pembayaran dengan tampilan transfer bank dsbnya.
7. LA' Kitchen memang terbaik.
8. Semangat memberikan yang terbaik yeey.
9. Membuat desain tampilan yang lebih menarik lagi.
10. Matul (Mantap betul).
11. Tambahkan rekomendasi resep makanan.
12. Baik.
13. Boleh menambahkan kemudahan lebih untuk pengguna, seperti ketika melakukan search (kata kuncinya tidak dihilangkan dari search bar), untuk pengguna baru (setelah melakukan update profile otomatis menuju ke halaman pembayaran). Overall sudah layak untuk dikembangkan.
14. Mantap.
15. Sistem sangat membantu dan mempercepat pekerjaan saya.

### 5.3.2. Kuesioner Penjual

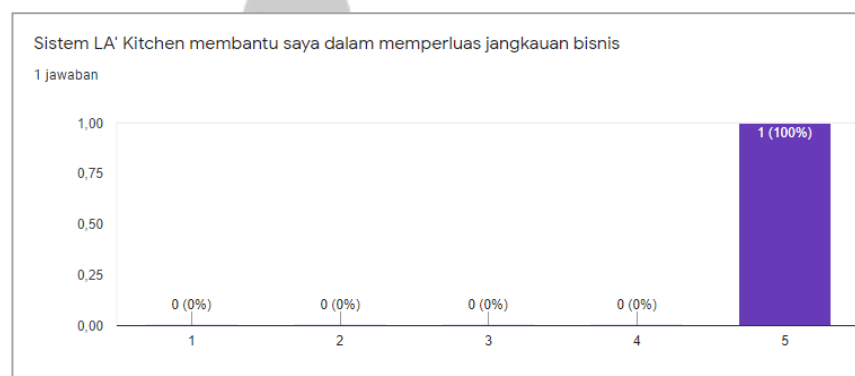
Kuesioner ini diperuntukkan secara khusus bagi penjual atau dengan kata lain pemilik LA' Kitchen itu sendiri. Berikut ini merupakan hasil skala terhadap pernyataan kuesioner dari penjual yang telah mencoba dan menggunakan sistem.

- 1) Seperti yang terlihat pada Gambar 5.36. responden menjawab sangat setuju bahwa sistem LA' Kitchen sangat membantu dalam melakukan pemasaran produk.



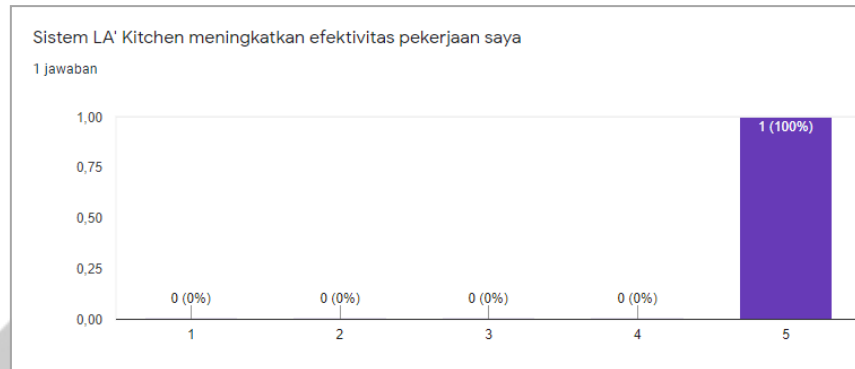
**Gambar 5.36. Kuesioner penjual pernyataan pertama.**

- 2) Seperti yang terlihat pada Gambar 5.37. responden menjawab sangat setuju bahwa sistem LA' Kitchen membantu penjual dalam memperluas jangkauan bisnis.



**Gambar 5.37. Kuesioner penjual pernyataan kedua.**

- 3) Seperti yang terlihat pada Gambar 5.38. responden menjawab sangat setuju bahwa sistem LA' Kitchen meningkatkan efektivitas pekerjaan pengguna.



**Gambar 5.38. Kuesioner penjual pernyataan ketiga,**

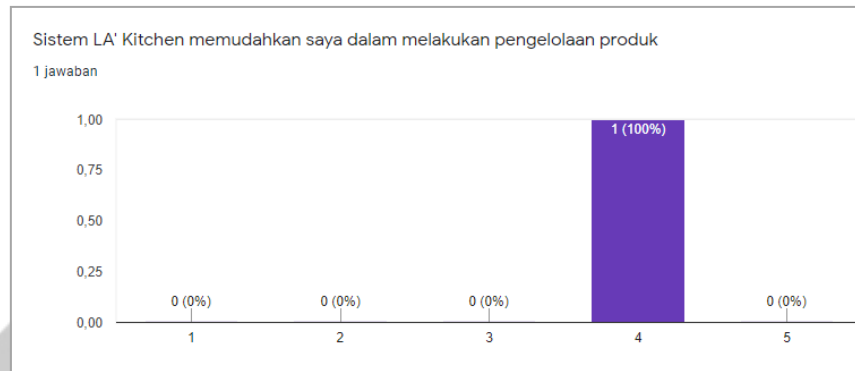
- 4) Seperti yang terlihat pada Gambar 5.39. responden menjawab sangat setuju bahwa sistem LA' Kitchen menyediakan otomatisasi laporan yang dibutuhkan.



**Gambar 5.39. Kuesioner penjual pernyataan keempat.**

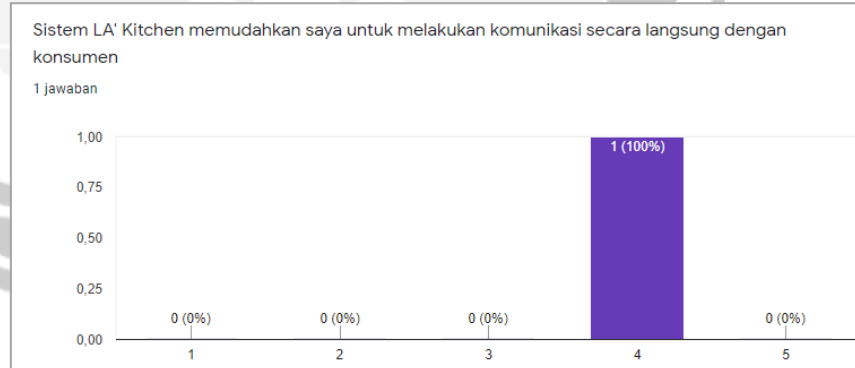


- 5) Seperti yang terlihat pada Gambar 5.40. responden menjawab setuju bahwa sistem LA' Kitchen memudahkan dalam melakukan pengelolaan produk.



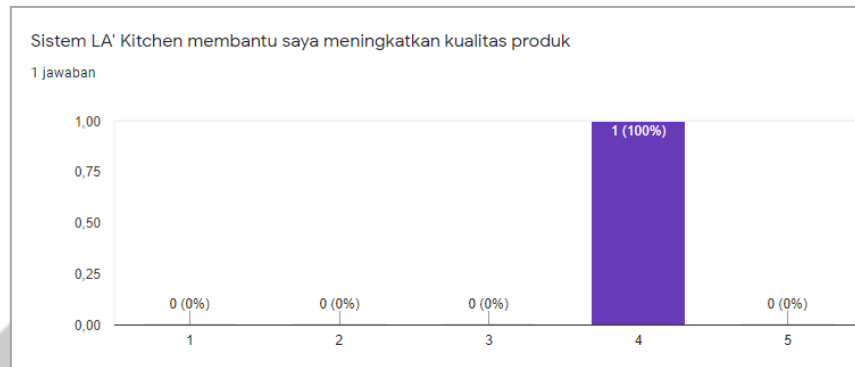
**Gambar 5.40. Kuesioner penjual pernyataan kelima.**

- 6) Seperti yang terlihat pada Gambar 5.41. responden menjawab setuju bahwa sistem LA' Kitchen memudahkan untuk melakukan komunikasi secara langsung dengan konsumen



**Gambar 5.41. Kuesioner penjual pernyataan keenam.**

- 7) Seperti yang terlihat pada Gambar 5.42. responden menjawab setuju bahwa sistem LA' Kitchen membantu dalam meningkatkan kualitas produk.



**Gambar 5.42. Kuesioner penjual pernyataan ketujuh.**

- 8) Seperti yang terlihat pada Gambar 5.43. responden menjawab sangat setuju bahwa sistem LA' Kitchen meningkatkan pertukaran komunikasi dengan konsumen.



**Gambar 5.43. Kuesioner penjual pernyataan kedelapan.**

Berdasarkan skala yang diberikan oleh pembeli dan penjual, terlihat bahwa semua responden memberikan respon yang positif terhadap pernyataan kuesioner sistem LA' Kitchen. Hal ini terbukti dengan rata-rata indeks penilaian sebesar 89,44% untuk kuesioner pembeli dan 92.5% untuk kuesioner penjual, yang menunjukkan dalam skala sangat setuju. Dari hasil pernyataan dan saran yang disampaikan responden, penulis menyimpulkan bahwa sistem LA' Kitchen masih memiliki beberapa kelebihan dan kekurangan.

Kelebihan :

- Sistem LA' Kitchen memudahkan dalam melakukan transaksi *online*.
- Sistem LA' Kitchen memiliki desain yang menarik dan enak dipandang.
- Sistem LA' Kitchen memiliki tampilan yang mudah dipahami.
- Sistem LA' Kitchen menyajikan informasi yang jelas dan cepat.
- Sistem LA' Kitchen meningkatkan efektivitas pekerjaan.
- Sistem LA' Kitchen membantu meningkatkan kualitas produk.

Kekurangan :

- Sistem LA' Kitchen memiliki metode pembayaran yang cukup rumit.
- Sistem LA' Kitchen memiliki beberapa simbol yang membingungkan bagi pengguna.
- Sistem LA' Kitchen masih memiliki beberapa tombol yang tidak berfungsi.