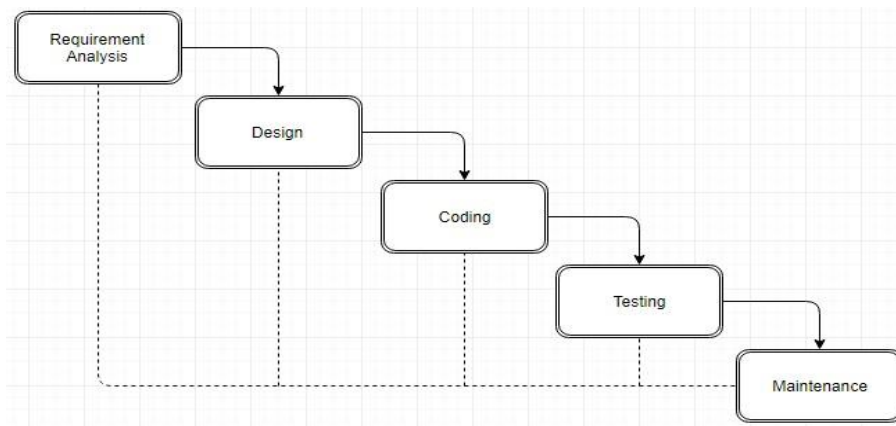


## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metodologi Penelitian

Penelitian tentang pengembangan aplikasi Dapur Salam berbasis *website* menggunakan metode *Software Development Life Cycle (SDLC)* model *waterfall* digambarkan seperti pada Gambar 3.1



**Gambar 3.1** SDLC Model *waterfall*

Berikut tahap-tahap pengembangan sistem berdasarkan metode *waterfall*:

*a. System atau Information Engineering and Modelling*

Tahap ini adalah tahap awal dari model SDLC *Waterfall*. pada tahap ini dilakukan pencarian kebutuhan dari keseluruhan system yang akan diaplikasikan dalam bentuk *software*. Pada tahap ini ditonjolkan bagaimana *software* bisa berinteraksi dengan element-element lain seperti *hardware*, *database*, dan lain-lain.

*b. Requirements Analysis*

Pada tahap ini difokuskan pencarian kebutuhan *software*. Pengembangan harus mengerti tentang domain informasi dari *software* agar mengetahui sifat dari system yang akan dibuat, seperti bagaimana *user interface* yang dibutuhkan, dan kebutuhan fungsi-fungsi lainnya pada *software*.

*c. Design*

Pada tahap ini pengembang akan mengubah kebutuhan-kebutuhan yang sudah didapat menjadi representasi dalam bentuk "*blueprint*" *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasi kebutuhan-kebutuhan yang sudah disebutkan pada tahap sebelumnya. Pada tahap ini diharuskan melakukan dokumentasi sebagai konfigurasi dari *software*.

*d. Coding*

Tahap ini adalah tahap yang sangat penting. Pada tahap ini pengembangan akan mengubah bentuk desain menjadi bentuk yang bias dimengerti oleh mesin, yaitu Bahasa pemrograman. Tahap ini merupakan implementasi dari tahap *design* yang dalam penelitian ini penulis mengerjakannya menggunakan bahasa PHP.

*e. Testing atau Verification*

Tahap ini adalah tahap dimana *software* hasil pembuatan harus diuji coba, termasuk semua fungsi-fungsinya. Proses ini bertujuan agar *software* yang dibuat terbebas dari masalah, dan hasilnya sesuai dengan kebutuhan-kebutuhan yang sudah didefinisikan sebelumnya.

*f. Maintenance*

Tahap akhir ini adalah tahap pemeliharaan *software*, dimana *software* yang telah dibuat memiliki tahap pemeliharaan atau pembaharuan. Pada tahap ini dimungkinkan penambahan fitur-fitur baru serta perbaikan jika terdapat masalah pada *system* yang dikembangkan. Pada penelitian ini penulis tidak mencapai tahap ini.

### **3.2 Tempat dan Waktu Penelitian**

Tempat yang digunakan penulis untuk melakukan penelitian ini yaitu Dapur Salam. Adapun waktu penelitian tentang pengembangan *website* Dapur Salam ini dilaksanakan dari bulan Juli sampai Agustus 2017.

### 3.3 Teknik Pengumpulan Data

Untuk memperoleh data yang diperlukan dalam penyusunan laporan Tugas Akhir ini, penelitian penulis melakukan beberapa tahapan sebagai berikut:

1. Menganalisis Kebutuhan

Analisis kebutuhan pada dasarnya merupakan tahap merancang dan membangun sebuah sistem informasi. Analisis kebutuhan merupakan suatu proses untuk mendapatkan informasi, model dan spesifikasi tentang perangkat lunak yang dibutuhkan pada aplikasi. Sebelum membuat program aplikasi berbasis *website*, terlebih dahulu melakukan wawancara kepada ketua pengurus di Dapur Salam yang bertujuan untuk mendapatkan data maupun informasi terkait bagaimana fitur-fitur yang diharapkan oleh Dapur Salam supaya *website* yang dibuat sesuai dengan harapan Dapur Salam.

2. Pengumpulan data dan menentukan kebutuhan

Pengumpulan data berasal dari *requirement* yang telah ditentukan berdasarkan penggabungan data primer dan sekunder. *Requirement* merupakan daftar kebutuhan dan persyaratan dari aplikasi. Dengan adanya *requirement*, pembuatan aplikasi akan dapat terarah dan terstruktur. *Requirement* juga dapat membantu dalam melakukan *testing* ketika aplikasi telah selesai dibuat.

### 3.4 Alat dan Bahan Penelitian

Penelitian membutuhkan alat-alat untuk mendukung berjalannya perancangan dan implementasi *website*, antara lain:

### 3.4.1 Perangkat Keras

#### *Laptop*

- a. *32 atau 64bit architecture processor*
- b. *4 GB Random Access Memory (RAM)*
- c. *Mouse*

### 3.4.2 Perangkat Lunak

#### a. Sistem Operasi *Windows 10*

*Windows 10* merupakan sistem operasi komputer pribadi yang dikembangkan oleh Microsoft sebagai bagian dari keluarga sistem operasi *windows NT*. *Windows 10* bertujuan untuk mengatasi kekurangan dalam antarmuka pengguna pertama kali diperkenalkan oleh *windows 8* dengan menambahkan mekanik tambahan yang dirancang untuk meningkatkan pengalaman pengguna untuk perangkat yang tidak ada layar sentuh (seperti komputer meja dan *laptop*), termasuk kebangkitan menu mulai yang terlihat di *windows 7*, sistem desktop maya, dan kemampuan untuk menjalankan aplikasi bursa *windows* dalam jendela pada desktop daripada modus layar penuh.

#### b. *Netbeans*

Perangkat lunak *Netbeans* merupakan alat utama dalam melaksanakan pembuatan *web* Dapur Salam.

#### c. MySQL

MySQL digunakan untuk membuat dan mengelola *database* beserta isinya.

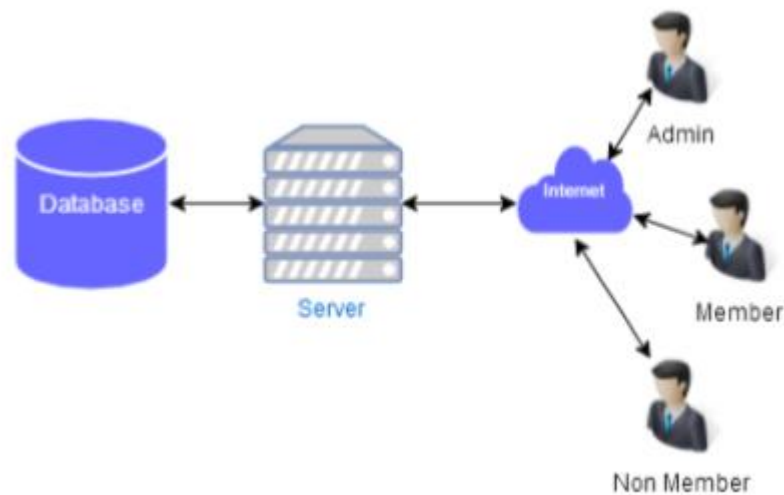
#### d. XAMPP

Sebagai alat yang digunakan untuk menjadi sebuah *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan Bahasa pemrograman *PHP* dan *Perl*.

#### e. Bootstrap

Sebagai alat yang digunakan untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain *website*.

#### 3.4.3 Arsitektur



**Gambar 3.2** Arsitektur Aplikasi

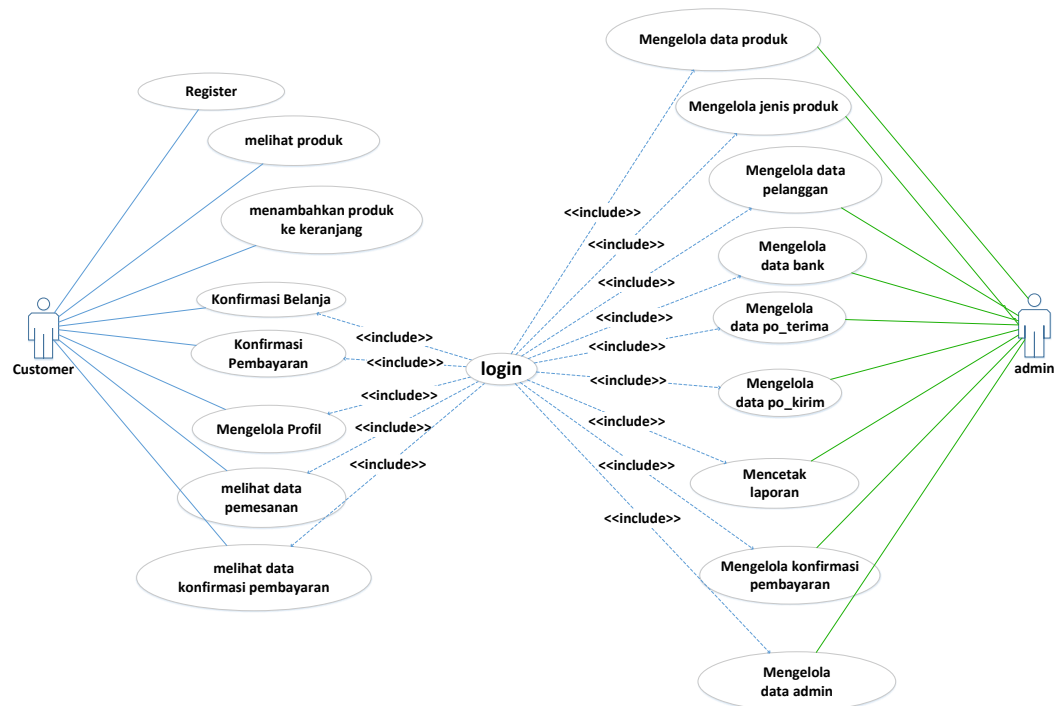
*Database server* menggunakan aplikasi MySQL dan menggunakan XAMPP sebagai *web server*. Hubungan antar pengguna dan *web server* menggunakan *internet* dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *web server* memuat antar muka dan melakukan pengambilan data yang dilakukan dari *database server*. Melalui antar muka yang dimuat *web server* sebagai pengguna bisa menyimpan *database server*.

#### 3.5 Rancangan Sistem dan Basis Data

Dalam pembuatan aplikasi dilakukan perancangan *database* menggunakan bantuan Diagram ER. Metode perancangan lain yang digunakan dalam aplikasi adalah *Unified Modeling Language (UML)* yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Activity Diagram*, *Class Diagram*.

### 3.5.1 Use Case Diagram

Gambaran *Use Case* Diagram yang digunakan dalam aplikasi dapat dilihat pada gambar 3.3



**Gambar 3.3** Use Case Diagram

Berdasarkan *Use Case* Gambar 3.3 dapat dilihat *actor* administrator, berikut penjelasannya.

#### a. Admin

*Admin* merupakan *actor* yang dapat melakukan aktifitas apapun yang terdapat di dalam *website*. *Admin* dapat diartikan seperti *super user* yang kegiatannya tidak dibatasi.

1. *Login*
2. Mengelola data produk
3. Mengelola jenis produk
4. Mengelola data pelanggan
5. Mengelola data bank

6. Mengelola data po terima
7. Mengelola data po kirim
8. Mengelola konfirmasi pembayaran
9. Mencetak laporan
10. Mengelola data *admin*

Semua kegiatan yang ada di *website admin* dapat digunakan oleh *admin*. *Admin* diberikan hak penuh untuk mengelolah *website admin*. Dalam *use case* diagram pada gambar 3.3 terdapat penghubung antar *use case* berupa relasi *association*.

*b. Customer*

Aktor *Customer* merupakan pengguna yang dapat menggunakan *website* namun hak aksesnya dibatasi oleh administrator. *Customer* hanya diberikan izin untuk melakukan beberapa hal. Ada beberapa kegiatan yang tidak dapat dilakukan oleh *customer*, namun ada juga yang dapat dilakukan oleh *customer*. Berikut daftar kegiatan yang dapat dilakukan oleh *customer*.

1. *Register*
2. *Login*
3. Melihat produk
4. Menambahkan produk ke keranjang
5. Konfirmasi belanja
6. Konfirmasi pembayaran
7. Mengelola profil
8. Melihat data pemesanan
9. Mengelola konfirmasi pembayaran

Dilihat dari daftar kegiatan yang dapat dilakukan oleh *customer*, dapat diartikan bahwa *customer* dapat mengelola konfirmasi belanja namun tidak bisa mengubah status konfirmasi.

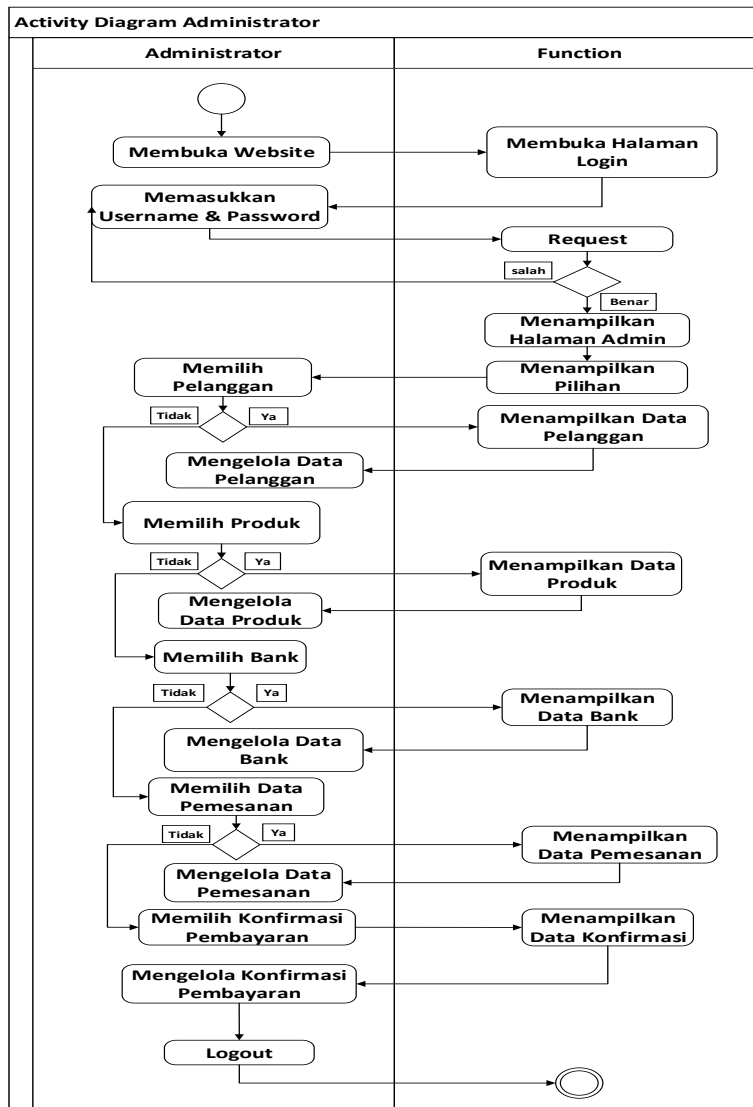
### 3.5.2 Activity Diagram

*Activity* Diagram akan menjelaskan bagaimana jalannya aplikasi dan hubungan antara pengguna, *website* dan *database*. *Activity* diagram pada rancangan *website* yang akan dibuat dibagi menjadi beberapa *Activity* diagram yaitu:

a. *Activity* Diagram Administrator

Gambar 3.4 menunjukkan *Activity* Diagram Administrator pada *web* administrator menggambarkan bagaimana kegiatan yang dilakukan oleh *user* administrator dengan *website*. *Website* yang dimaksud adalah *website* pengolah *website* utama. Administrator setelah membuka *website* harus mengisi *form login*. Jika *username* dan *password* yang dimasukkan oleh administrator adalah benar, maka administrator akan masuk kedalam menu halaman utama yang banyak terdapat pilihan-pilihan kegiatan yang dapat dilakukan oleh administrator. Namun jika *username* dan *password* yang dimasukkan oleh administrator adalah salah, maka *website* akan memberikan notifikasi bahwa *username* dan *password* yang dimasukkan oleh administrator salah dan tetap berada di halaman *login*. Jika sudah masuk, administrator akan diberikan beberapa pilihan yang dapat dipilih. Pilihan yang dipilih oleh administrator secara langsung akan direspon oleh *website* untuk menampilkan halaman terkait dengan data yang ada.



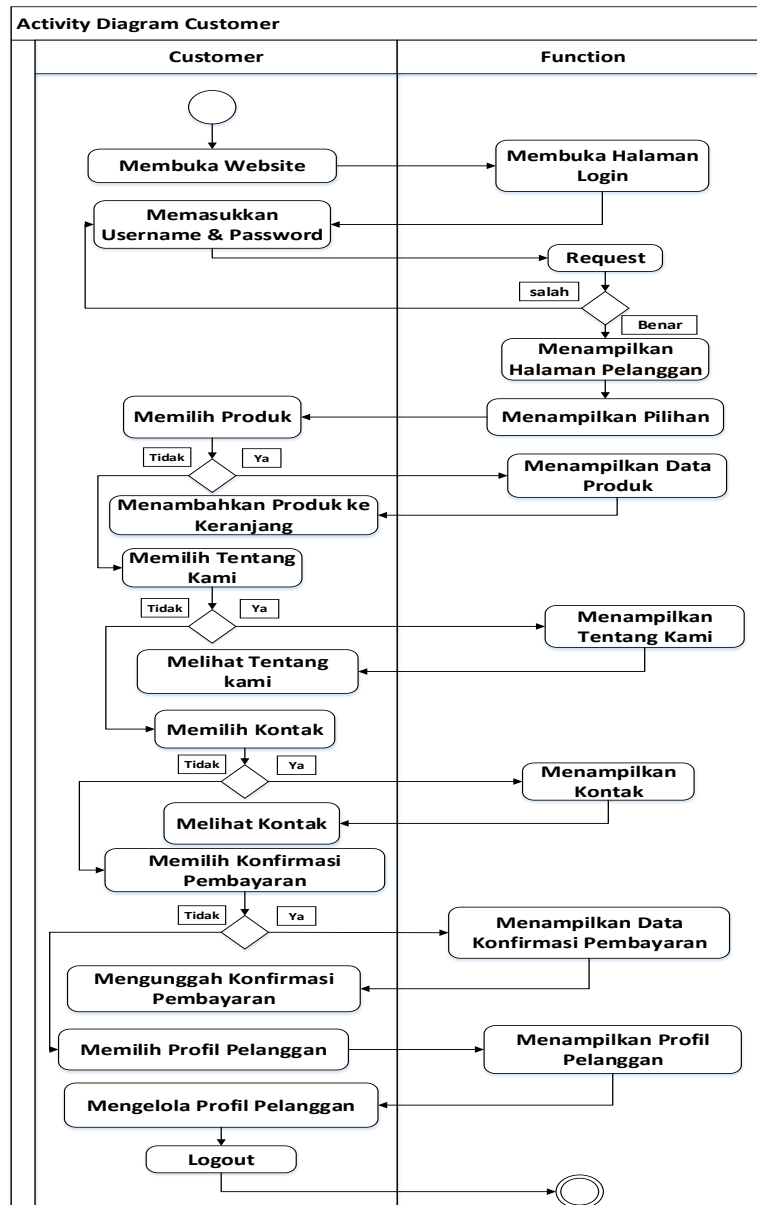


**Gambar 3.4** Activity Diagram Administrator

*b. Activity Diagram Customer*

Gambar 3.5 menunjukkan Activity Diagram Customer pada *web customer* menggambarkan bagaimana kegiatan yang dilakukan oleh *user customer* dengan *website*. *Website* yang dimaksud adalah *website* pengolah *website customer*. *Customer* setelah membuka *website* harus mengisi *form login*. Jika *username* dan *password* yang dimasukkan oleh *customer* adalah benar, maka *customer* akan masuk kedalam menu halaman utama yang sama seperti *user public* namun di bagian halaman keranjang *customer* dapat melanjutkan pembayaran. Namun jika

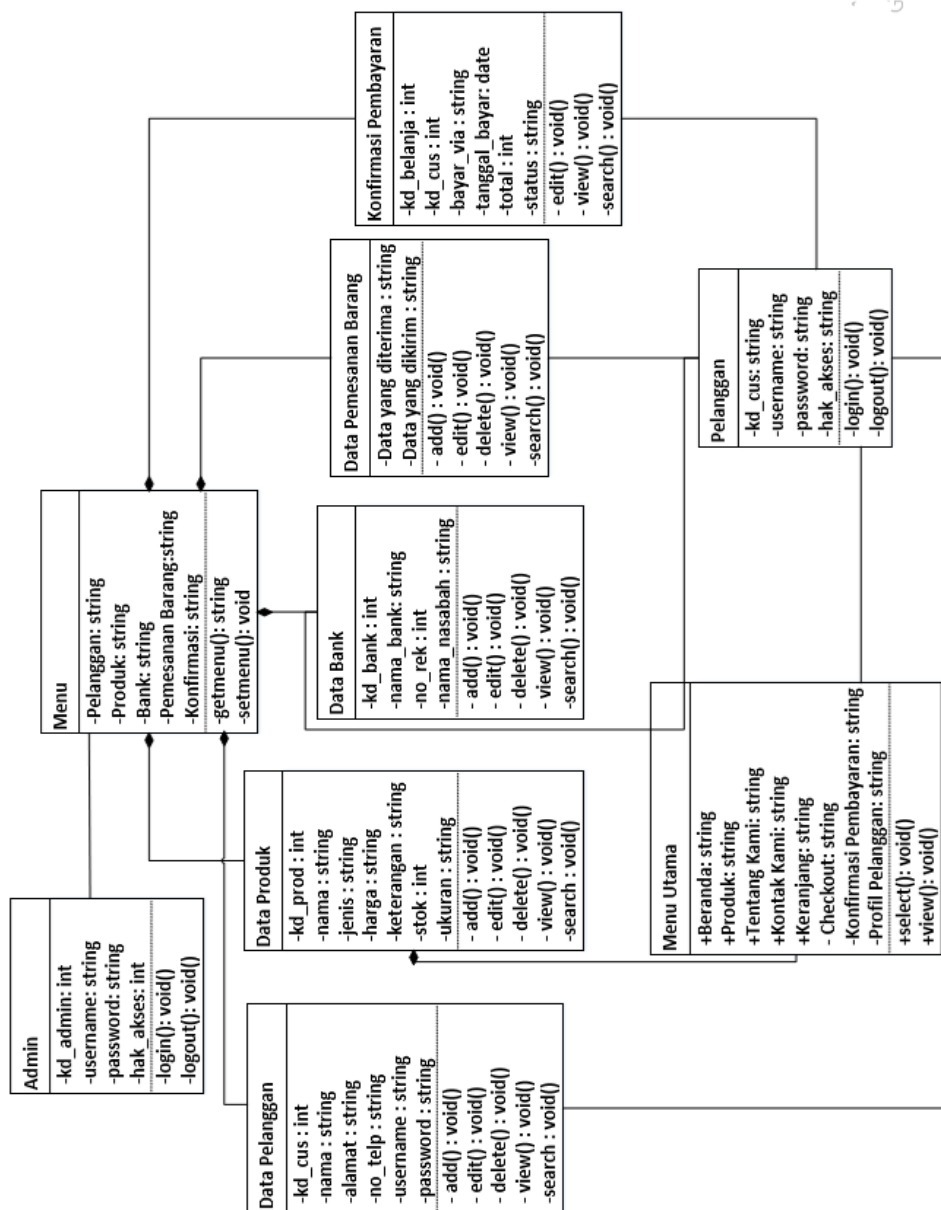
*username* dan *password* yang dimasukkan oleh *customer* adalah salah, maka *website* akan memberikan notifikasi bahwa *username* dan *password* yang dimasukkan oleh *customer* salah dan tetap berada di halaman *login*.



**Gambar 3.5** Activity Diagram Customer

### 3.5.3 Class Diagram

Class diagram dibuat untuk menjelaskan hubungan antara kelas yang satu dengan kelas yang lainnya. Di dalam beberapa kelas, terdapat atribut yang menjelaskan komponen apa saja yang terdapat di dalam kelas tersebut dan terdapat *operation* yang berfungsi sebagai penjabar kegiatan apa saja yang dapat dilakukan oleh suatu aktor ketika berada di dalam kelas tersebut. Gambaran Class Diagram yang digunakan dalam aplikasi dapat dilihat pada gambar Gambar 3.6.



Gambar 3.6 Class Diagram

Berikut adalah penjelasan *class* diagram pada gambar 3.6

- a. Pada *class* menu utama, memiliki fungsi untuk menampilkan menu yang dapat dipilih oleh pelanggan dan *public*.
- b. Pada *class* menu, memiliki fungsi untuk menampilkan menu yang dapat dipilih oleh administrator.
- c. Pada *class* data pelanggan, memiliki fungsi untuk mengelola data semua pelanggan, di dalam *class* data pelanggan admin bisa mengedit, mencari, menampilkan, dan menghapus.
- d. Pada *class* data produk, memiliki fungsi untuk mengelola semua data produk, di dalam *class* data produk admin bisa menambahkan, mencari, mengedit, menampilkan dan menghapus.
- e. Pada *class* data bank, memiliki fungsi untuk mengelola semua data bank, di dalam *class* data bank admin bisa menambahkan, mencari, mengedit, menampilkan dan menghapus.
- f. Pada *class* data pemesanan barang, admin memiliki fungsi untuk mengelola data yang diterima dan data yang dikirim.
- g. Pada *class* konfirmasi pembayaran, memiliki fungsi untuk mengelola semua konfirmasi pembayaran, di dalam *class* konfirmasi pembayaran admin dapat mengedit dan mencari.
- h. Pada *class* admin, memiliki fungsi untuk masuk ke menu.
- i. Pada *class* pelanggan, memiliki fungsi untuk masuk ke menu utama

*Class* admin memiliki *association* dengan *class* menu, artinya *class* admin berhak mengakses apa saja yang telah tersedia di dalam *class* menu melalui *method* *getmenu()*.

*Class* menu memiliki *compotition* dengan *class* data pelanggan. Relasi tersebut berarti *class* data pelanggan merupakan komponen penyusun dari *class* menu. Dengan relasi *compotition*, tergambar jelas bahwa *class* data pelanggan tidak dapat berdiri sendiri untuk digunakan melainkan harus terlebih dahulu berasal dari *class* menu.

*Class* menu memiliki *composition* dengan *class* data produk. Relasi tersebut berarti *class* data produk merupakan komponen penyusun dari *class* menu. Dengan relasi *composition*, tergambar jelas bahwa *class* data produk tidak dapat berdiri sendiri untuk digunakan melainkan harus terlebih dahulu berasal dari *class* menu.

*Class* menu memiliki *composition* dengan *class* data bank. Relasi tersebut berarti *class* data bank merupakan komponen penyusun dari *class* menu. Dengan relasi *composition*, tergambar jelas bahwa *class* data bank tidak dapat berdiri sendiri untuk digunakan melainkan harus terlebih dahulu berasal dari *class* menu.

*Class* menu memiliki *composition* dengan *class* data pemesanan barang. Relasi tersebut berarti *class* data pemesanan barang merupakan komponen penyusun dari *class* menu. Dengan relasi *composition*, tergambar jelas bahwa *class* pemesanan barang tidak dapat berdiri sendiri untuk digunakan melainkan harus terlebih dahulu berasal dari *class* menu.

*Class* menu memiliki *composition* dengan *class* data konfirmasi pembayaran. Relasi tersebut berarti *class* data konfirmasi pembayaran merupakan komponen penyusun dari *class* menu. Dengan relasi *composition*, tergambar jelas bahwa *class* konfirmasi pembayaran tidak dapat berdiri sendiri untuk digunakan melainkan harus terlebih dahulu berasal dari *class* menu.

*Class* pelanggan memiliki *association* dengan *class* data bank, artinya pelanggan boleh melihat data bank melalui method *view()*.

*Class* pelanggan memiliki *association* dengan *class* data pelanggan, artinya pelanggan boleh mengubah data pelanggan tersebut melalui method *edit()* dan melihat data pelanggan tersebut melalui method *view()*.

*Class* pelanggan memiliki *association* dengan *class* data pemesanan barang, artinya pelanggan boleh menambahkan pemesanan ke data yang diterima melalui method *add()*, mengubah data pemesanan melalui method *edit()*, dan melihat data pemesanan yang telah dipesan pelanggan tersebut melalui method *view()*.

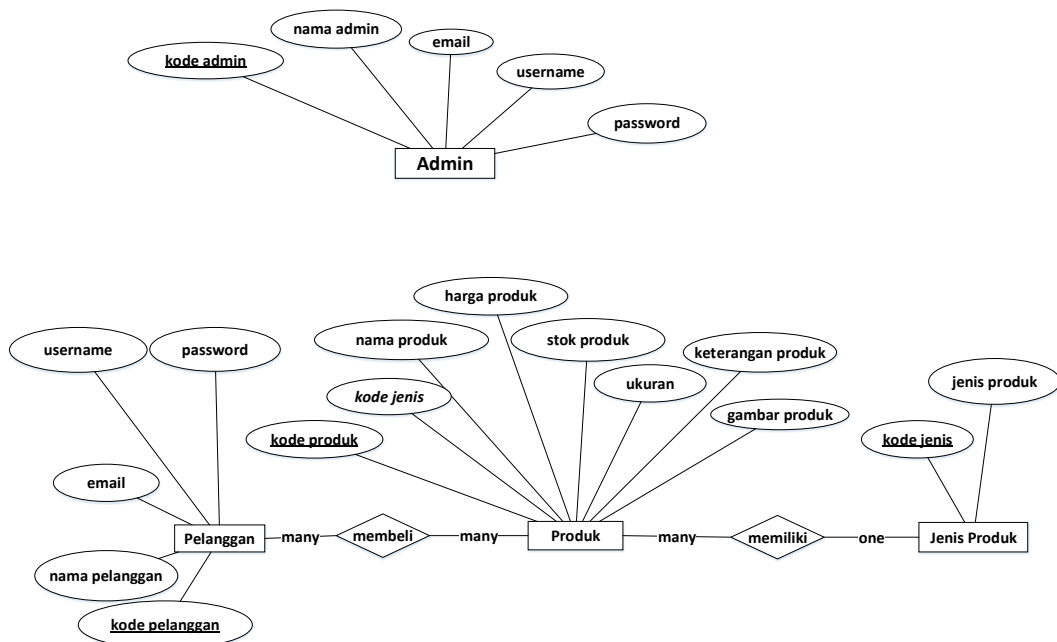
*Class* pelanggan memiliki *association* dengan *class* konfirmasi pembayaran, artinya pelanggan boleh melihat data konfirmasi melalui method *view()*, mengubah data konfirmasi melalui method *edit()*.

*Class* pelanggan memiliki *association* dengan *class* menu utama, artinya pelanggan dapat memilih dan melihat beberapa menu yang hanya dapat diakses oleh pelanggan, seperti *checkout*, konfirmasi pembayaran, dan profil.

*Class* menu utama memiliki *compositition* dengan *class* data produk, artinya data produk merupakan *class* penyusun menu utama. *Class* data produk tidak bisa ditampilkan apabila tidak difungsikan oleh *class* menu utama.

### 3.5.4 Entity Relationship Diagram

ER Diagram merupakan diagram yang digunakan untuk merancang basis data dan memperlihatkan relasi antar entitas atau objek dengan atribut. Hasil pada tahap desain *Entity Relationship* Diagram dalam aplikasi *website* dapat dilihat pada gambar 3.7.



**Gambar 3.7** Entity Relationship Diagram

Pada gambar 3.7 dapat dilihat bahwa *entity relationship* diagram yang dirancang memiliki 4 buah entitas yaitu:

- a. Pelanggan
- b. Produk
- c. Jenis Produk
- d. *Admin*

Berikut penjelasan *entity relationship* diagram pada gambar 3.6

Entitas pelanggan memiliki relasi dengan produk artinya setiap pelanggan dapat melakukan pemesanan produk. Entitas pelanggan memiliki relasi dengan transaksi artinya pelanggan dapat melakukan transaksi. Transaksi akan menghasilkan beberapa tabel baru sehingga pelanggan lebih mudah dalam melakukan transaksi. Entitas pelanggan memiliki relasi dengan *po\_terima* artinya setiap pelanggan dapat memesan produk. Entitas pelanggan memiliki relasi dengan *dataongkir* artinya setiap pelanggan yang telah memesan akan menerima data ongkos kirim. Entitas pelanggan memiliki relasi dengan *konfirmasi* artinya pelanggan yang sudah melakukan pemesanan dan mendapatkan data ongkos kirim diharuskan untuk melakukan konfirmasi sehingga proses transaksi dapat dilanjutkan.

Entitas produk memiliki relasi dengan entitas jenis produk artinya setiap produk memiliki jenis produk. Entitas produk memiliki relasi dengan entitas *po\_terima* artinya setiap produk dapat dipesan.

Entitas *Admin* tidak memiliki relasi dengan entitas lain sehingga dapat disimpulkan entitas *admin* dapat berdiri sendiri dan tidak bergantung pada entitas lain.

### **3.5.5 Konversi ER Diagram ke dalam Tabel**

Konversi ERD ke dalam bentuk tabel adalah proses memasukan entitas menjadi tabel dan atribut menjadi kolom beserta relasinya ke dalam sebuah tabel yang berhubungan dengan tabel lainnya.





**Tabel 3.5** Tabel Transaksi

<b>kd_pt</b>	<b>kd_cus</b>	<b>kd_prod</b>	<b>no belanja</b>	<b>tanggal terima</b>	<b>jumlah barang</b>	<b>sub total</b>

Entitas pelanggan memiliki relasi *many to many* dengan entitas produk sehingga menghasilkan tabel transaksi. Kode pelanggan (kd\_cus) dan kode produk (kd\_prod) merupakan *primary key* dari tabel pelanggan dan produk. Kode pelanggan (kd\_cus) dan kode produk (kd\_prod) yang terdapat pada tabel transaksi menjadi *foreign key*. Di dalam tabel transaksi juga terdapat kolom nobelanja, tanggalterima, jumlahbarang, dan subtotal.

**Tabel 3.6** Transaksi

<b>kd_pt</b>	<b>kd_cus</b>	<b>kd_prod</b>	<b>kd_ong</b>	<b>kd_kon</b>	<b>kd_pk</b>	<b>kd_bank</b>	<b>no belanja</b>	<b>tanggal terima</b>	<b>jumlah barang</b>	<b>sub total</b>	<b>ala mat</b>	<b>prov</b>	<b>kab</b>	<b>kurir</b>	<b>Jenis kurir</b>

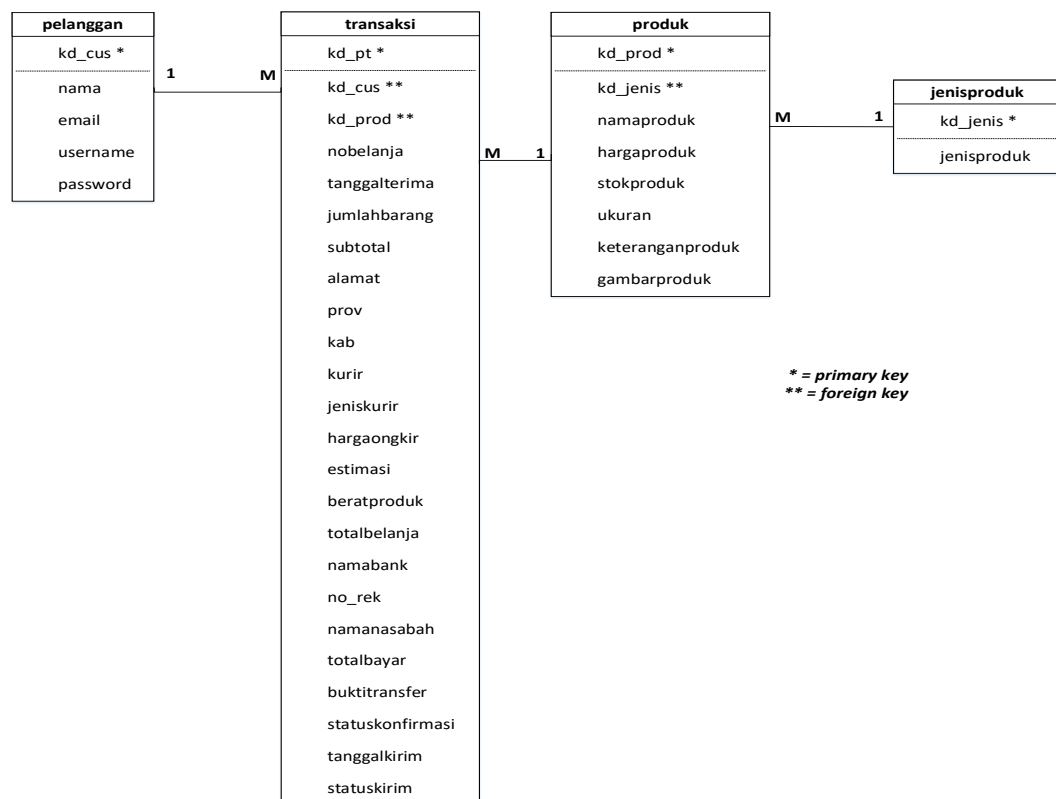
<b>harga ongkir</b>	<b>esti masi</b>	<b>berat produk</b>	<b>total belanja</b>	<b>nama bank</b>	<b>no rek</b>	<b>nama nasabah</b>	<b>total bayar</b>	<b>bukti transfer</b>	<b>status konfir</b>	<b>tanggal kirim</b>	<b>status kirim</b>

Tabel 3.6 transaksi merupakan hasil relasi antara entitas pelanggan dengan entitas produk. Di dalam tabel transaksi ada kolom kd\_pt, kd\_cus, kd\_prod, nobelanja, tanggalterima, jumlah barang dan subtotal. Ada juga beberapa kolom yang ditambahkan dalam tabel transaksi untuk proses transaksi yaitu:

- kd\_ong, adalah kode ongkos kirim.
- kd\_kon, adalah kode konfirmasi pembayaran.
- kd\_pk, adalah kode produk order yang dikirim.
- kd\_bank, adalah kode dari akun bank.
- alamat, adalah alamat dari pelanggan.
- provinsi, adalah provinsi dari pelanggan.
- kabupaten, adalah kabupaten dari pelanggan.
- kurir, adalah jasa pengiriman produk.
- jeniskurir, adalah jenis kurir dari jasa pengiriman produk.
- hargaongkir, adalah harga ongkos kirim dari transaksi

- estimasi, adalah perkiraan hari produk akan sampai di tempat pelanggan.
- beratproduk, adalah berat dari produk yang dipesan.
- totalbelanja, adalah total penjumlahan dari sub total dan harga ongkir.
- namabank, adalah nama dari bank yang ada pada transaksi.
- no\_rek, adalah nomor rekening dari bank.
- totalbayar, adalah total yang harus dibayar oleh pelanggan saat konfirmasi pembayaran.
- buktitransfer, adalah bukti pembayaran apabila pelanggan sudah melakukan pembayaran.
- statuskonfirmasi, adalah status dari konfirmasi pembayaran.
- tanggalkirim, adalah tanggal kirimnya produk ke pelanggan.
- statuskirim, adalah status dari pengiriman produk ke pelanggan.

Hasil konversi ER Diagram beserta relasi antara tabel dapat dilihat pada gambar 3.8



**Gambar 3.8** Relasi Antar Tabel

### 3.5.6 Normalisasi

Normalisasi adalah teknik untuk mengorganisir hasil konversi erd ke dalam tabel sehingga dapat digunakan dengan baik. Normalisasi menghilangkan data yang tidak konsisten dan juga memperbaiki tabel yang menyimpang. Gambar 3.9 adalah hasil konversi erd yang perlu dinormalisasi sehingga dapat digunakan dengan efisien.

kd_pt	kd_cus	kd_prod	kd_ong	kd_kon	kd_pk	kd_bank	no_belanja	tanggal_terima	jumlah_barang	sub_total	ala_mat	prov	kab	kurir	Jenis_kurir

harga_ongkir	esti_masi	berat_produk	total_belanja	nama_bank	no_rek	nama_nasabah	total_bayar	bukti_transfer	status_konfir	tanggal_kirim	status_kirim

**Gambar 3.9** Normalisasi Tabel Transaksi

Pada gambar 3.9 menunjukkan hasil normalisasi tabel transaksi yang merupakan hasil konversi *entity relationship* diagram (erd). Hasil normalisasi pada tabel transaksi menghasilkan rancangan tabel po\_terima, dataongkir, konfirmasi, po\_kirim, dan akunbank.

**Tabel 3.7** Tabel po\_terima

kd_pt	kd_cus	kd_prod	nobelanja	tanggal_terima	Jumlah_barang	subtotal

**Tabel 3.8** Tabel dataongkir

Kd_ongkir	kd_cus	no_belanja	ala_mat	prov	kab	kurir	jenis_kurir	Harga_ongkit	esti_masi	Berat_produk	Total_belanja

**Tabel 3.9** Tabel Konfirmasi

Kd_kon	Kd_cus	nobelanja	kd_bank	totalbayar	buktitransfer	status

Tabel 3.10 Tabel akunbank

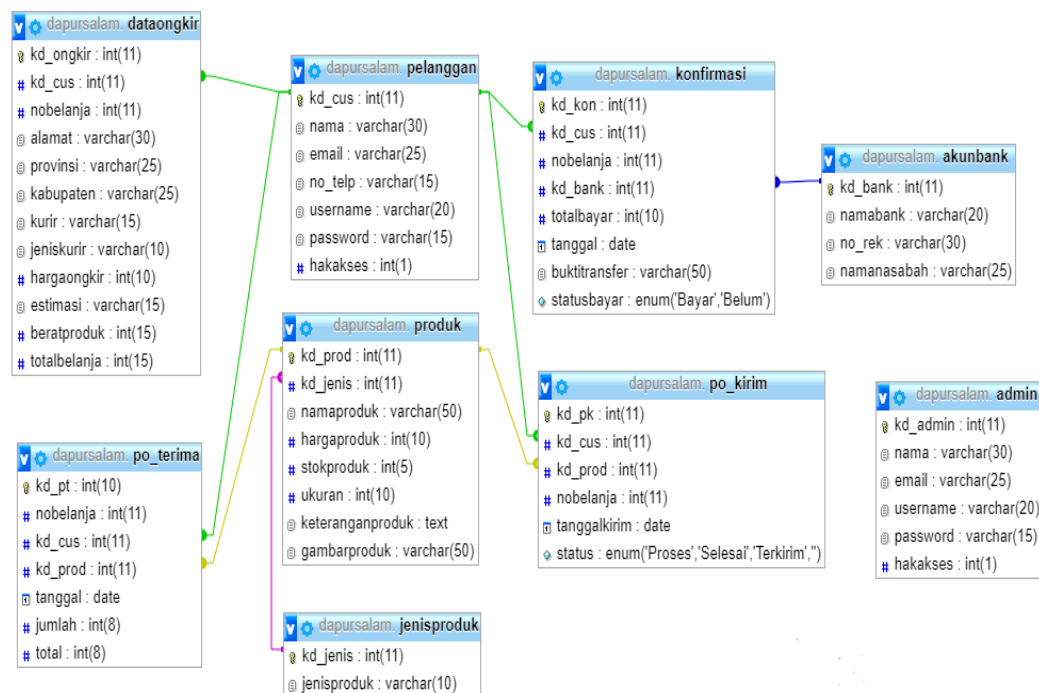
kd_bank	namabank	no_rek	namanasabah

Tabel 3.11 Tabel Po\_kirim

kd_pk	kd_cus	kd_prod	nobelanja	tanggal kirim	status kirim

### 3.5.7 Relasi Antar Tabel

Rancangan tabel yang dibuat sebelumnya telah melalui proses normalisasi sehingga dapat diimplementasikan ke dalam basis data tanpa adanya data yang tidak konsisten, rangkap, dan menyimpang. Tabel hasil rancangan telah diimplementasikan dan menghasilkan relasi antar tabel seperti pada gambar 3.10.



Gambar 3.10 Relasi Antar Tabel

### 3.5.8 Kamus Data

**Tabel 3.12** Struktur Tabel Pelanggan

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	Kd_cus	Int	Auto Increment	pk	Kode Pelanggan merupakan <i>primary key</i> dari tabel pelanggan
2	Nama	Varchar	30		Nama pelanggan berupa nama dari pelanggan yang mendaftarkan akun sebagai member
3	Email	Varchar	25		Email pelanggan berupa alamat email dari pelanggan
4	No_telp	Varchar	15		Nomor <i>handphone</i> pelanggan.
5	Username	Varchar	20		<i>Username</i> pelanggan sebagai nama pengguna untuk login ke halaman pelanggan
6	Password	Varchar	15		<i>Password</i> pelanggan sebagai password untuk <i>login</i> ke halaman pelanggan

**Tabel 3.13** Struktur Tabel Produk

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	kd_prod	Int	Auto Increment	pk	Kode Produk merupakan <i>primary key</i> dari tabel produk
2	kd_jenis	Int	11	fk	Kode jenis diambil dari tabel jenisproduk
3	Namaproduk	Varchar	50		Nama sebuah produk
4	Hargaproduk	int	10		Harga sebuah produk
5	Stokproduk	Int	5		Stok sebuah produk
6	Ukuran	Int	10		Ukuran berat sebuah produk
7	Keteranganproduk	Text			Keterangan sebuah produk
8	Gambarproduk	Varchar	50		Gambar sebuah produk

**Tabel 3.14** Struktur Tabel JenisProduk

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	Kd_jenis	Int	Auto increment	pk	Kode jenis produk merupakan <i>primary key</i> dari tabel jenisproduk
2	Jenisproduk	Varchar	10		Nama jenis produk

**Tabel 3.15** Struktur Tabel po\_terima

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	Kd_pt	Int	Auto increment	pk	Kode Po Terima merupakan <i>primary key</i> dari tabel po_terima
2	Nobelanja	Int	11		Nomor belanja sebagai nomor pembelian produk di waktu pemesanan yang sama
3	Kd_cus	Int	11	fk	Kode pelanggan diambil dari tabel pelanggan
4	Kd_prod	Int	11	fk	Kode produk diambil dari tabel produk
5	Tanggal	Date			Tanggal pemesanan sebuah produk
6	Jumlah	Int	8		Jumlah barang yang dipesan
7	Subtotal	Int	8		Harga subtotal dari setiap pemesanan

**Tabel 3.16** Struktur Tabel dataongkir

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	Kd_ongkir	Int	Auto Increment	pk	Kode Ongkir merupakan <i>primary key</i> dari tabel dataongkir
2	Kd_cus	Int	11	fk	Kode Pelanggan diambil dari tabel pelanggan
3	Nobelanja	Int	11		Nomor belanja sebagai nomor pembelian produk di waktu pemesanan yang sama
4	Alamat	varchar	30		Alamat tujuan
5	Provinsi	varchar	25		Provinsi tujuan
6	Kabupaten	varchar	25		Kabupaten tujuan
7	Kurir	varchar	15		Kurir yang dipakai
8	JenisKurir	varchar	10		Jenis Kurir yang dipakai
9	Hargaongkir	Int	10		Harga ongkos kirim
10	Estimasi	varchar	15		Perkiraan durasi waktu pengiriman
11	Beratproduk	Int	15		Berat dari sebuah produk
12	Totalbelanja	Int	15		Total pembayaran dari produk yang dipesan ditambah harga ongkos kirim



**Tabel 3.17** Struktur Tabel konfirmasi

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	kd_kon	Int	Auto Increment	pk	Kode konfirmasi merupakan <i>primary key</i> dari tabel konfirmasi
2	kd_cus	Int	11	fk	Kode pelanggan diambil dari tabel pelanggan
3	nobelanja	Int	11		Nomor belanja sebagai nomor pembelian produk di waktu pemesanan yang sama
5	Kd_bank	int	11	fk	Kode bank diambil dari tabel akunbank agar pelanggan dapat memilih akun mana yang mau ditransfer.
4	totalbayar	Int	10		Total pembayaran
6	tanggal	Date			Tanggal konfirmasi pembayaran
7	buktitransfer	varchar	50		Gambar bukti pembayaran
8	status	enum			Menyediakan status bayar atau belum bayar

**Tabel 3.18** AkunBank

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	kd_bank	Int	Auto Increment	pk	Kode bank merupakan <i>primary key</i> dari tabel akunbank
2	Namabank	Varchar	20		Nama Bank adalah nama dari akun bank Dapur Salam
3	No_rek	Int	30		Nomor rekening dari akun bank Dapur Salam
4	Namanasabah	Varchar	25		Nama Nasabah dari akun bank Dapur Salam

**Tabel 3.19** Admin

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	Kd_admin	Int	Auto Increment	pk	Kode <i>admin</i> merupakan <i>primary key</i> dari tabel <i>admin</i>
2	Nama	Varchar	30		Nama <i>admin</i> dari Dapur Salam
3	Email	Varchar	25		Email <i>admin</i> dari Dapur Salam
4	Username	Varchar	20		<i>Username admin</i> sebagai nama pengguna untuk login ke halaman admin
5	Password	Varchar	15		<i>Password admin</i> sebagai password untuk <i>login</i> ke halaman <i>admin</i>

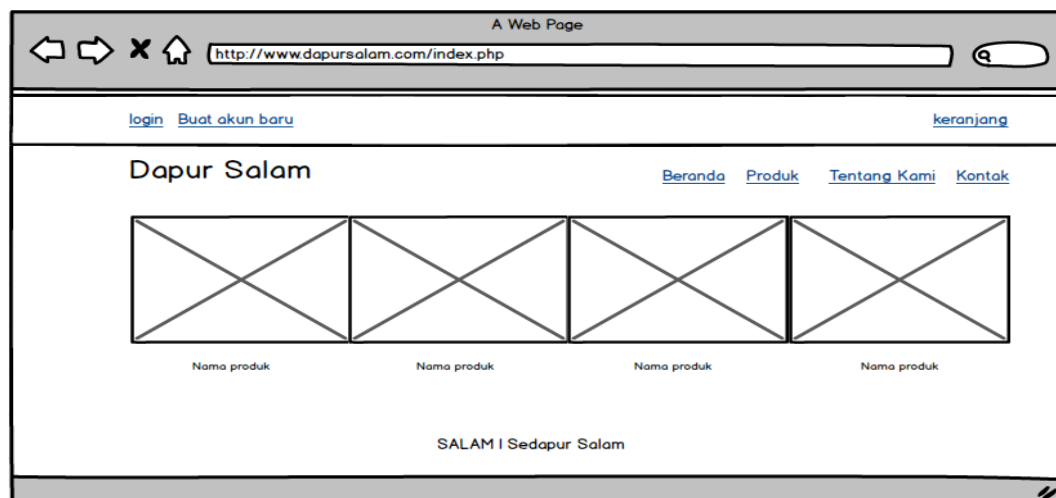
Tabel 3.20 Po\_Kirim

No	Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
1	kd_pk	Int	Auto Increment	Pk	Kode Po Kirim merupakan <i>primary key</i> dari tabel po_kirim
2	kd_cus	Int	11	fk	Kode pelanggan diambil dari tabel pelanggan
3	Kd_prod	Int	11	fk	Kode produk diambil dari tabel produk
4	nobelanja	Int	11		Nomor belanja sebagai nomor pembelian produk di waktu pemesanan yang sama
4	Tanggalkirim	Date			Tanggal dikirimnya produk ke pelanggan
5	Status	Enum			Menyediakan status proses, terkirim, selesai.

### 3.6 Rancangan Antar Muka

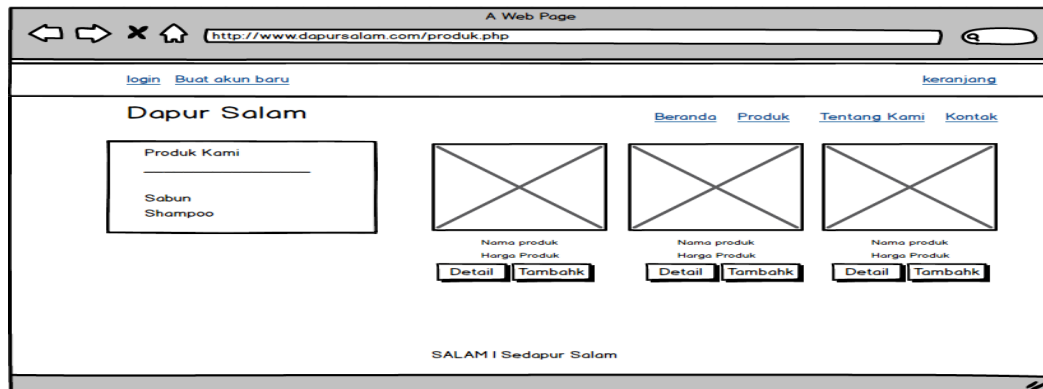
#### 3.6.1 Rancangan antarmuka halaman beranda *public*

Rancangan antarmuka halaman utama *public* merupakan tampilan utama dari *website* yang dapat dilihat oleh *public*. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.11.

Gambar 3.11 Rancangan Halaman Utama *Public*

### 3.6.2 Rancangan antarmuka halaman produk (*public*)

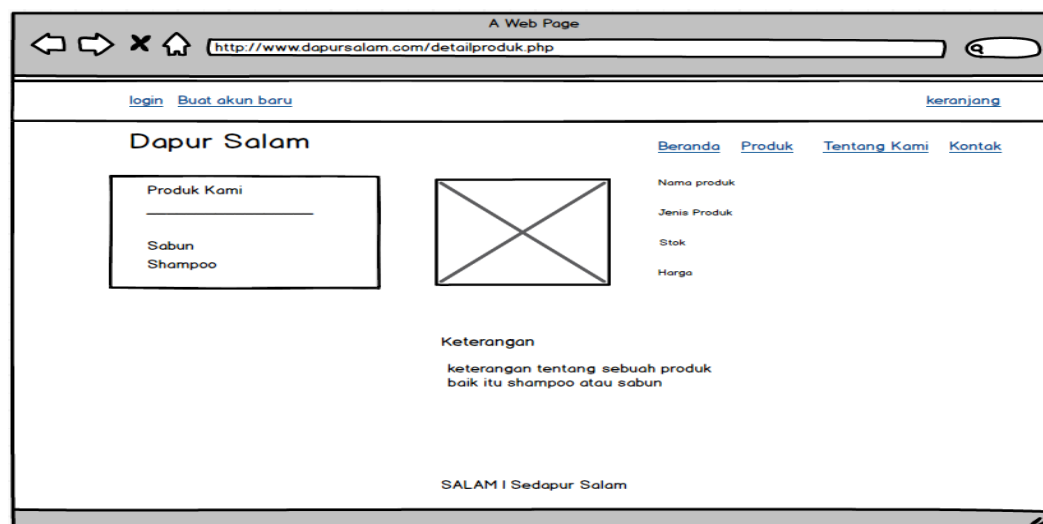
Rancangan antarmuka halaman produk menampilkan produk-produk yang dijual oleh Dapur Salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.12.



Gambar 3.12 Rancangan Halaman Produk

### 3.6.3 Rancangan antarmuka halaman detail produk

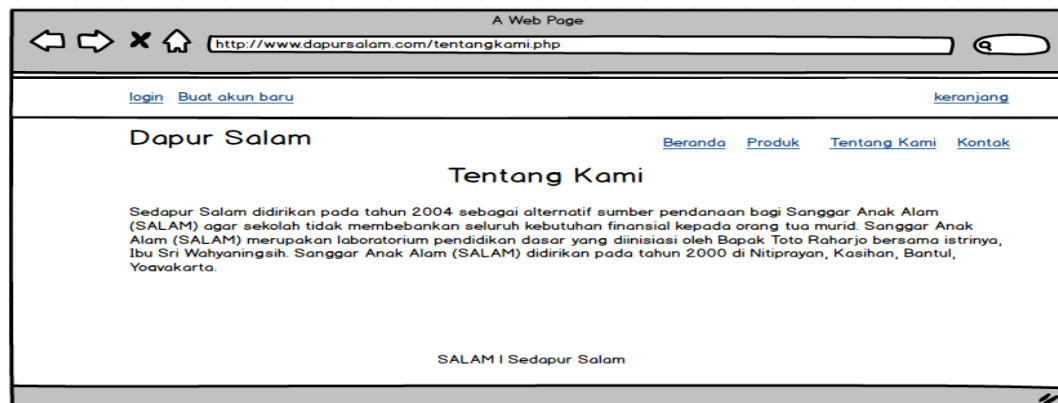
Rancangan antarmuka halaman detail produk menampilkan detail dari produk yang dipilih di halaman produk. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.13.



Gambar 3.13 Rancangan Halaman Detail Produk

### 3.6.4 Rancangan antarmuka halaman tentang kami

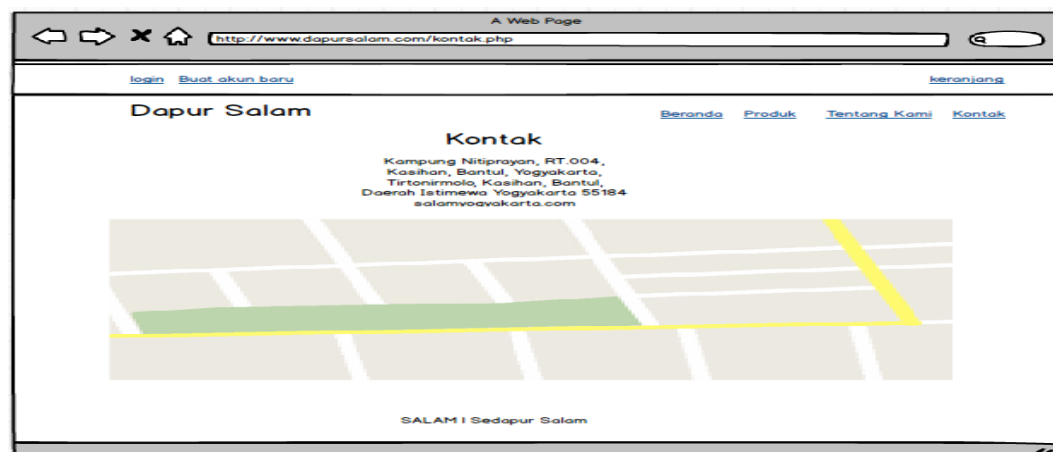
Rancangan antarmuka halaman tentang kami menampilkan penjelasan tentang awal dapur salam berdiri. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.14.



Gambar 3.14 Rancangan Halaman Tentang Kami

### 3.6.5 Rancangan antarmuka halaman kontak

Rancangan antarmuka halaman kontak menampilkan penjelasan tentang alamat, link *website* Salam Yogyakarta, dan *map* lokasi dapur salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.15.



Gambar 3.15 Rancangan Halaman Kontak

### 3.6.6 Rancangan antarmuka halaman daftar akun

Rancangan antarmuka halaman daftar akun merupakan halaman yang digunakan untuk mendaftarkan akun pelanggan. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.16.

The screenshot shows a web browser window with the URL <http://www.dapursalam.com/akun.php>. The page header includes navigation links: [login](#), [Buat akun baru](#), and [keranjang](#). The main content area features the 'Dapur Salam' logo and a navigation menu with [Beranda](#), [Produk](#), [Tentang Kami](#), and [Kontak](#). Below the logo is a registration form with the following fields: Nama, Email, Alamat, Nomor Handphone, Username, and Password. A 'Daftar' button is positioned below the Password field. At the bottom of the form area, the text 'SALAM | Sedapur Salam' is displayed.

Gambar 3.16 Halaman Daftar Akun

### 3.6.7 Rancangan antarmuka halaman *login*

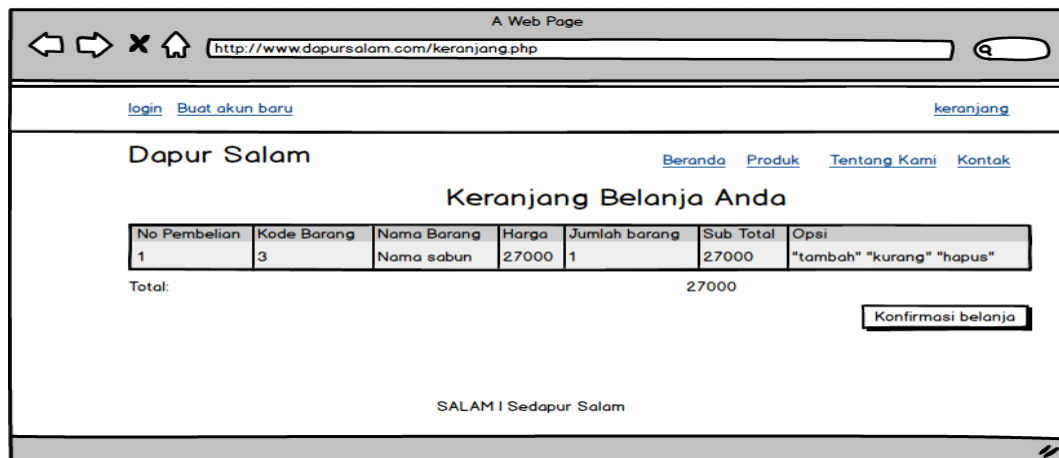
Rancangan antarmuka halaman *login* merupakan halaman yang berfungsi untuk dapat masuk ke halaman utama administrator atau pelanggan. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.17.

The screenshot shows a web browser window with the URL <http://www.dapursalam.com/login.php>. The page header includes navigation links: [login](#), [Buat akun baru](#), and [keranjang](#). The main content area features the 'Dapur Salam' logo and a navigation menu with [Beranda](#), [Produk](#), [Tentang Kami](#), and [Kontak](#). Below the logo, there are two sections: 'Pelanggan Baru' with a 'Buat Akun Baru' button, and 'Pelanggan yang telah terdaftar' with fields for Username and Password, and a 'Login' button. At the bottom of the form area, the text 'SALAM | Sedapur Salam' is displayed.

Gambar 3.17 Rancangan Halaman *Login*

### 3.6.8 Rancangan antarmuka halaman keranjang

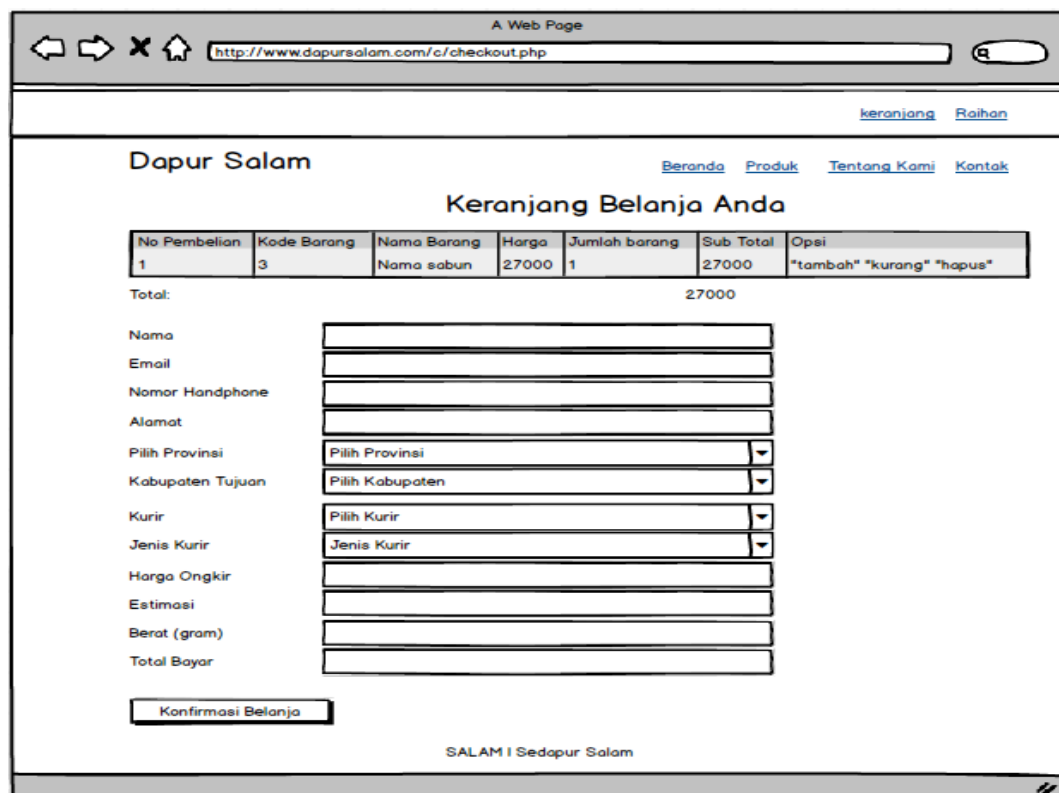
Rancangan antarmuka halaman tentang kami menampilkan tentang produk yang mau dibeli dan akan melakukan konfirmasi belanja, tetapi sebelum ke halaman *checkout*, pelanggan diwajibkan *login*. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.18.



Gambar 3.18 Rancangan Halaman Keranjang

### 3.6.9 Rancangan antarmuka halaman *checkout*

Rancangan antarmuka halaman *checkout* menampilkan produk yang mau dibeli, data-data yang harus diisi untuk dikonfirmasi pembelajarannya. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.19.



Gambar 3.19 Rancangan Halaman *Checkout*

### 3.6.10 Rancangan antarmuka halaman konfirmasi

Rancangan antarmuka halaman konfirmasi menampilkan produk yang udah dipesan dan data-data yang telah diisi. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.20.

Kode belanja	Nama Produk	Tanggal	Jumlah Produk	Sub Total	Tools
1	Sabun ini	2017-07-01	1	27000	"Batalkan Belanja"

N Pelanggan	Alamat	Provinsi	Kab	kurir	jj kurir	ongkir	estimasi	berat	total yang dibayar
Raihan	Belakang Kampus	DIY Yogyakarta	Gamping	JNE	OKE	20000	4-5	250	47000

Kode Belanja	Kode Pelanggan	Bayar Via	Tanggal	Jumlah	Status	Tools
1	15	-	2017-07-01	Rp. 00	Belum Bayar	"Konfirmasi Pembayaran"

SALAM | Sedapur Salam

Gambar 3.20 Rancangan Halaman Konfirmasi

### 3.6.11 Rancangan antarmuka halaman konfirmasi pembayaran

Rancangan antarmuka halaman konfirmasi pembayaran berfungsi untuk dapat menkonfirmasi pembayaran sesuai produk yang dibeli. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.21.

Keranjang Belanja Anda

Total yang harus dibayar: 27000

Kode Konfirmasi:

Metode Bayar:

Tanggal Bayar:

Jumlah:

Bukti Transfer:

Status: Belum Bayar

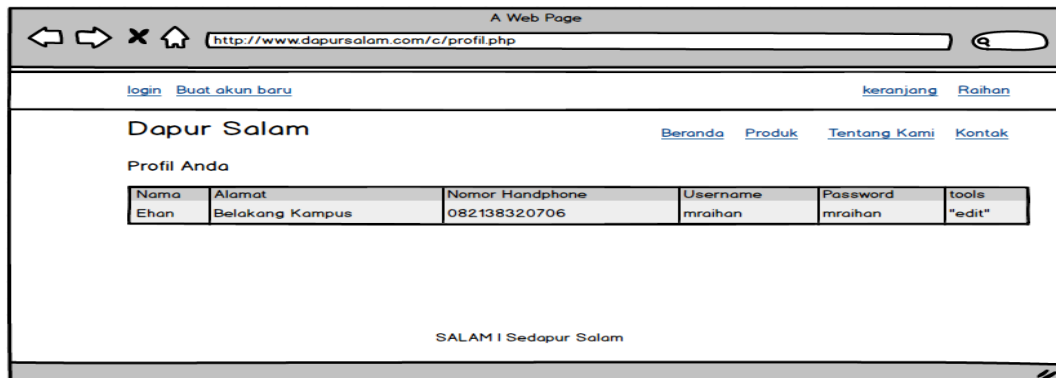
SALAM | Sedapur Salam

Gambar 3.21 Rancangan Halaman Konfirmasi Pembayaran



### 3.6.12 Rancangan antarmuka halaman profil pelanggan

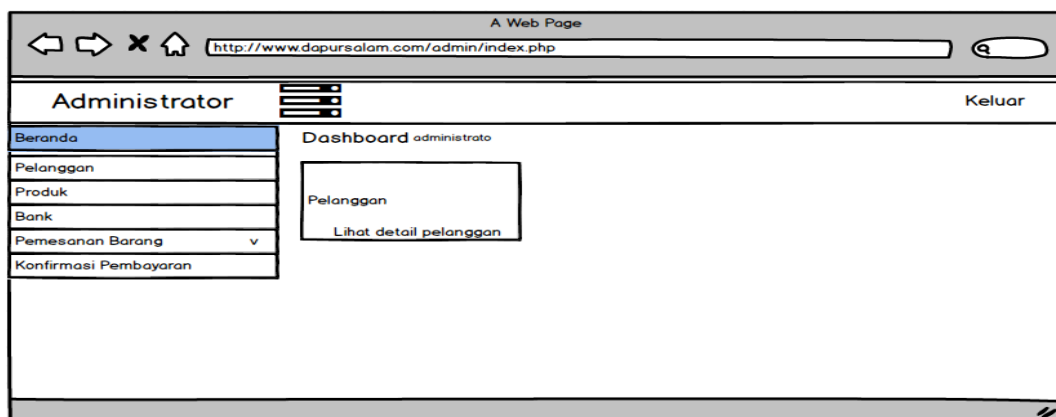
Rancangan antarmuka halaman profil pelanggan merupakan tampilan profil yang bisa diubah data pelanggan oleh pelanggan tersebut. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.22.



Gambar 3.22 Rancangan Halaman Profil Pelanggan

### 3.6.13 Rancangan antarmuka halaman beranda administrator

Rancangan antarmuka halaman utama administrator merupakan tampilan admin dari *website* yang hanya dapat dilihat oleh administrator. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.23.

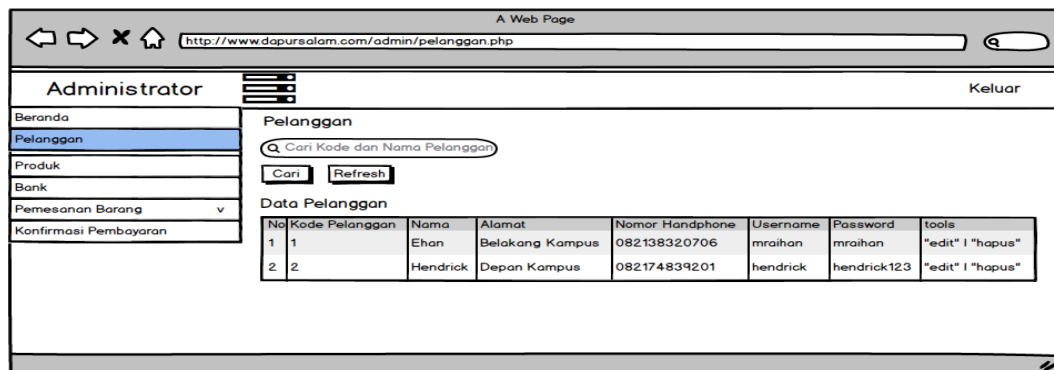


Gambar 3.23 Rancangan Halaman Beranda Administrator

### 3.6.14 Rancangan antarmuka halaman data pelanggan administrator

Rancangan antarmuka halaman pelanggan administrator merupakan tampilan data-data pelanggan, data pelanggan tersebut bisa diedit tapi tidak bisa

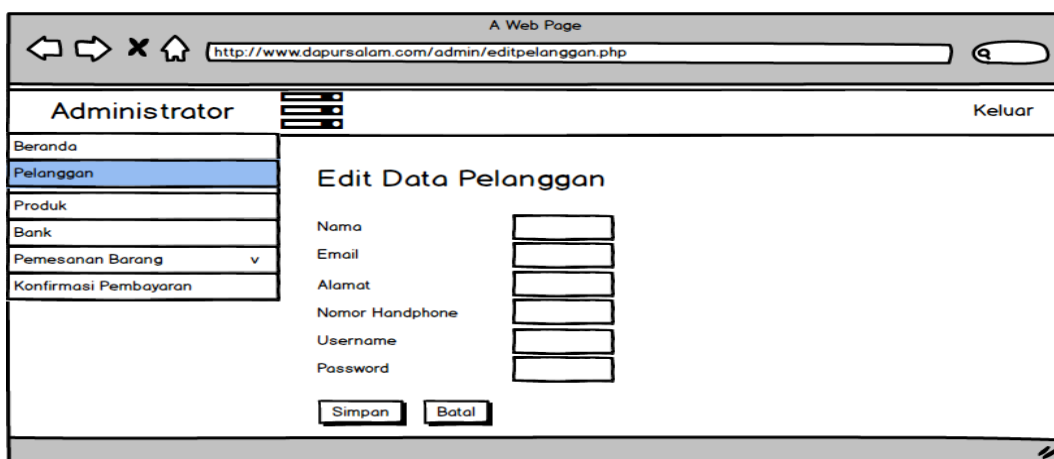
dihapus. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.24.



**Gambar 3.24** Rancangan Halaman Data Pelanggan Administrator

### 3.6.15 Rancangan antarmuka halaman edit pelanggan administrator

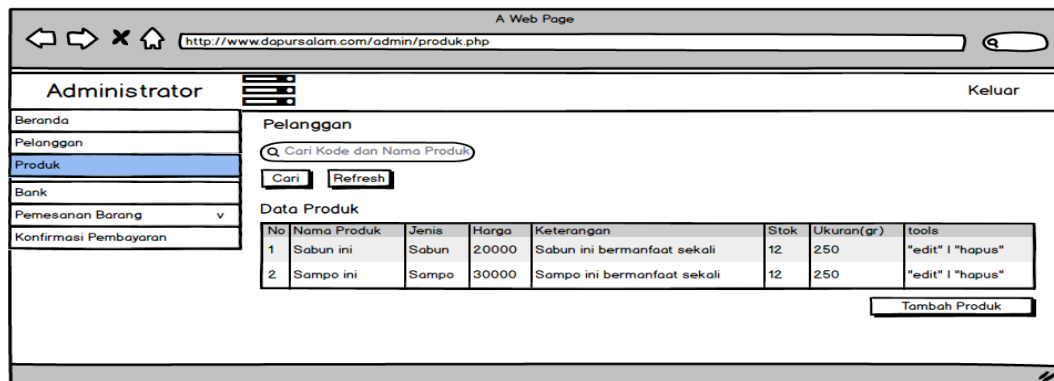
Rancangan antarmuka halaman edit pelanggan administrator berfungsi untuk memperbarui data pelanggan. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.25.



**Gambar 3.25** Rancangan Halaman Edit Pelanggan Administrator

### 3.6.16 Rancangan antarmuka halaman produk administrator

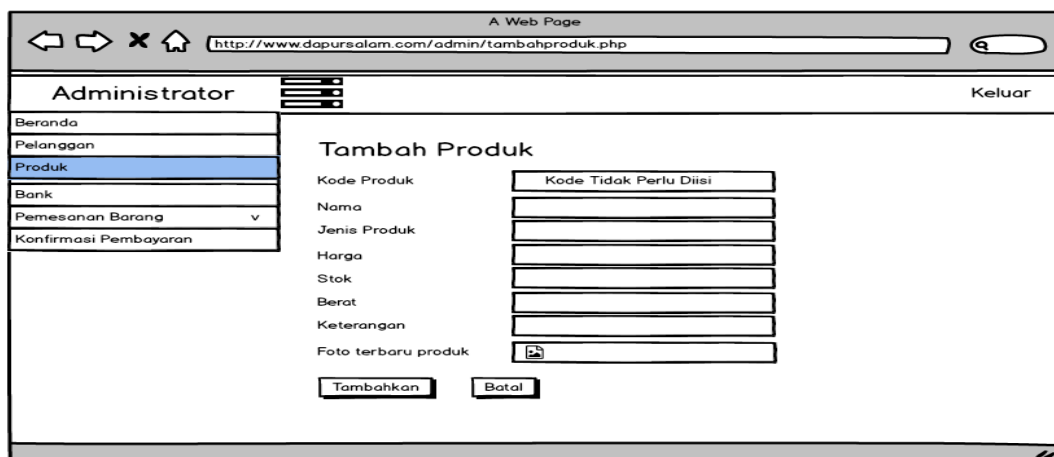
Rancangan antarmuka halaman produk administrator menampilkan data-data produk yang dijual di Dapur Salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.26.



**Gambar 3.26** Rancangan Halaman Produk Administrator

### 3.6.17 Rancangan antarmuka halaman tambah produk administrator

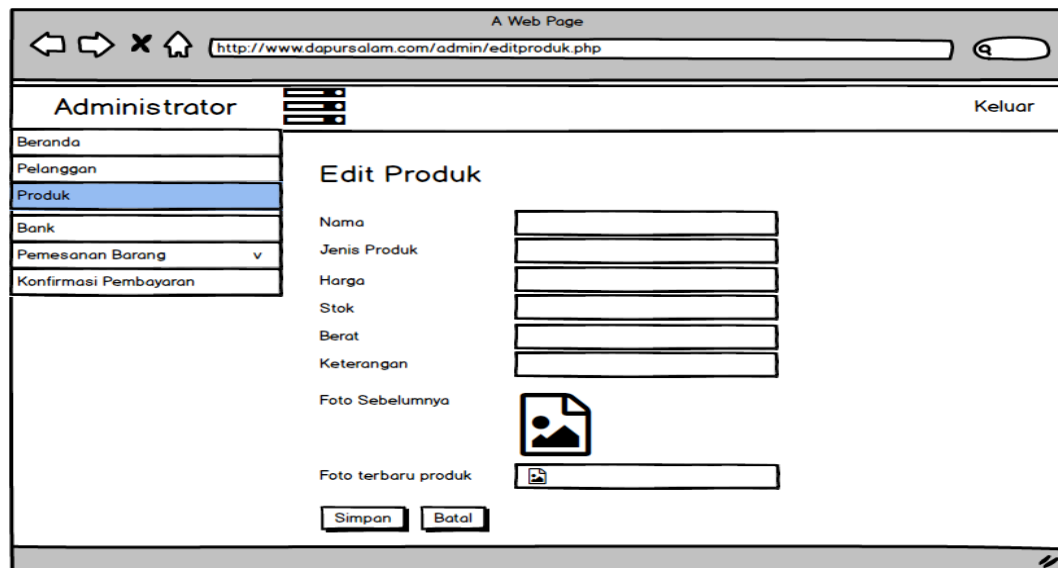
Rancangan antarmuka halaman tambah produk administrator berfungsi untuk menambahkan produk yang dijual di Dapur Salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.27.



**Gambar 3.27** Rancangan Halaman Tambah Produk

### 3.6.18 Rancangan antarmuka halaman edit produk administrator

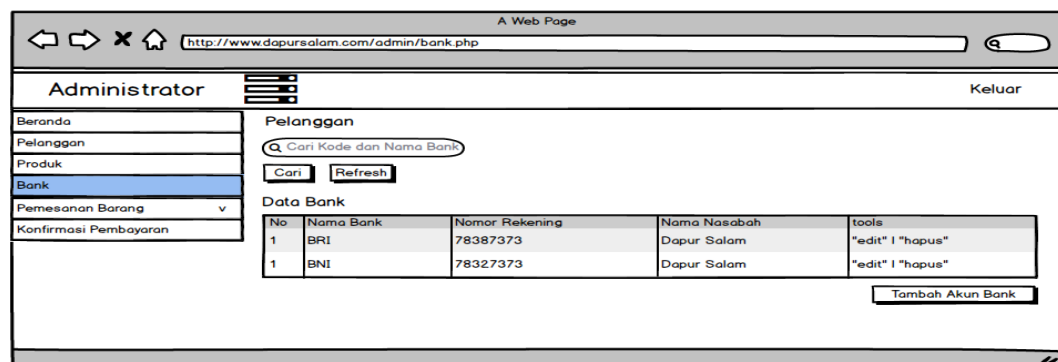
Rancangan antarmuka halaman edit produk administrator berfungsi untuk memperbarui data-data produk apabila ada kesalahan dalam menambahkan produk. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.28.



**Gambar 3.28** Rancangan Halaman Edit Produk

### 3.6.19 Rancangan antarmuka halaman bank administrator

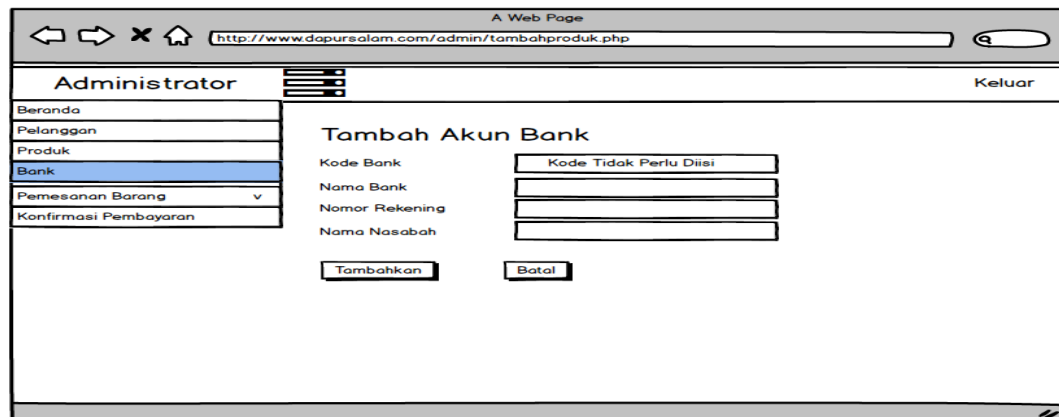
Rancangan antarmuka halaman produk administrator menampilkan data-data akun bank Dapur Salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.29.



**Gambar 3.29** Rancangan Halaman Bank

### 3.6.20 Rancangan antarmuka halaman tambah bank administrator

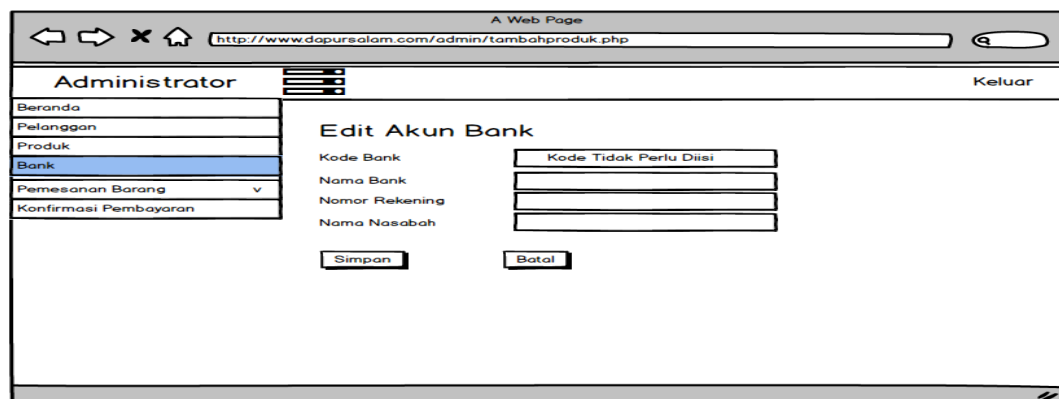
Rancangan antarmuka halaman tambah bank administrator berfungsi untuk menambahkan akun bank Dapur Salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.30.



**Gambar 3.30** Rancangan Halaman Tambah Akun Bank

### 3.6.21 Rancangan antarmuka halaman edit bank administrator

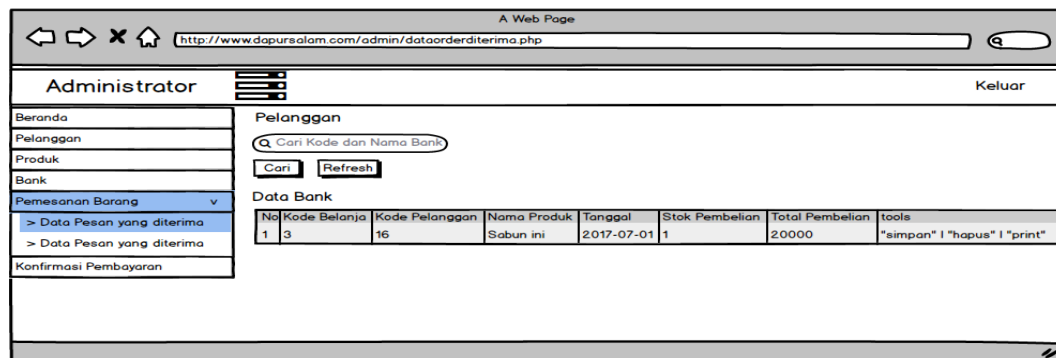
Rancangan antarmuka halaman edit bank administrator berfungsi untuk memperbarui data akun bank apabila ada kesalahan dalam menambahkan akun bank tersebut. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.31.



**Gambar 3.31** Rancangan Halaman Edit Bank

### 3.6.22 Rancangan antarmuka halaman data order diterima administrator

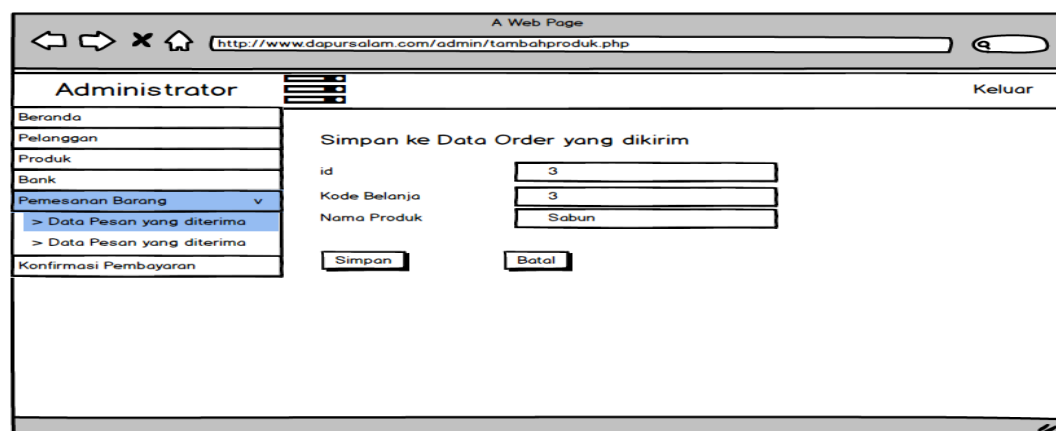
Rancangan antarmuka halaman data order diterima administrator menampilkan data-data produk dan pelanggan yang telah memesan produk di Dapur Salam. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.32.



**Gambar 3.32** Rancangan Halaman Data Order Diterima

### 3.6.23 Rancangan antarmuka halaman simpan order administrator

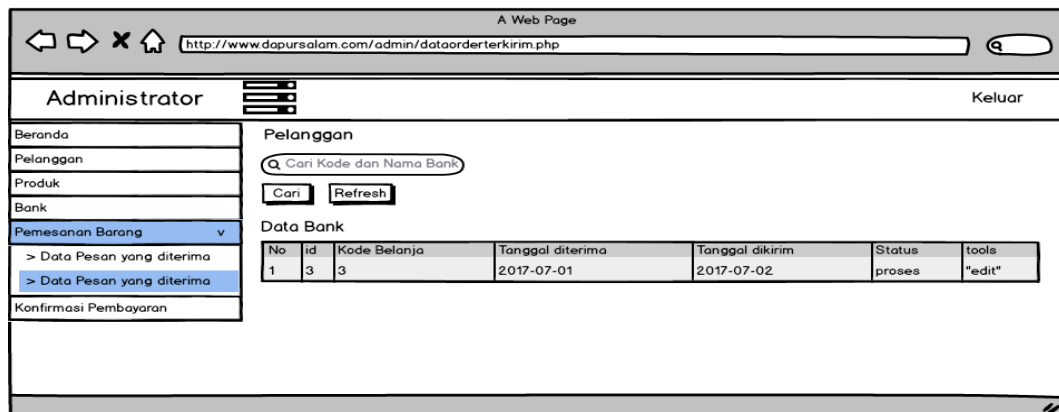
Rancangan antarmuka halaman simpan order administrator berfungsi untuk menyimpan pesanan yang berada di data order diterima ke halaman data order terkirim. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.33.



**Gambar 3.33** Rancangan Halaman Simpan Order ke Data Order Terkirim

### 3.6.24 Rancangan antarmuka halaman data order terkirim administrator

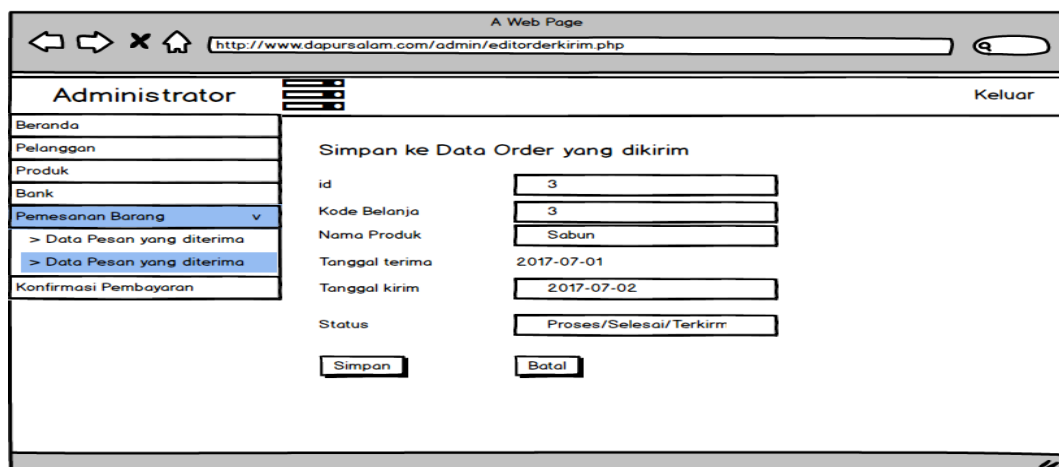
Rancangan antarmuka halaman data order terkirim menampilkan data-data yang telah dipesan dan diproses, kemudian data yang ada akan dikirim ke pelanggan apabila sudah dibayar. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.34.



Gambar 3.34 Rancangan Halaman Data Order Terkirim

### 3.6.25 Rancangan antarmuka halaman edit order diterima administrator

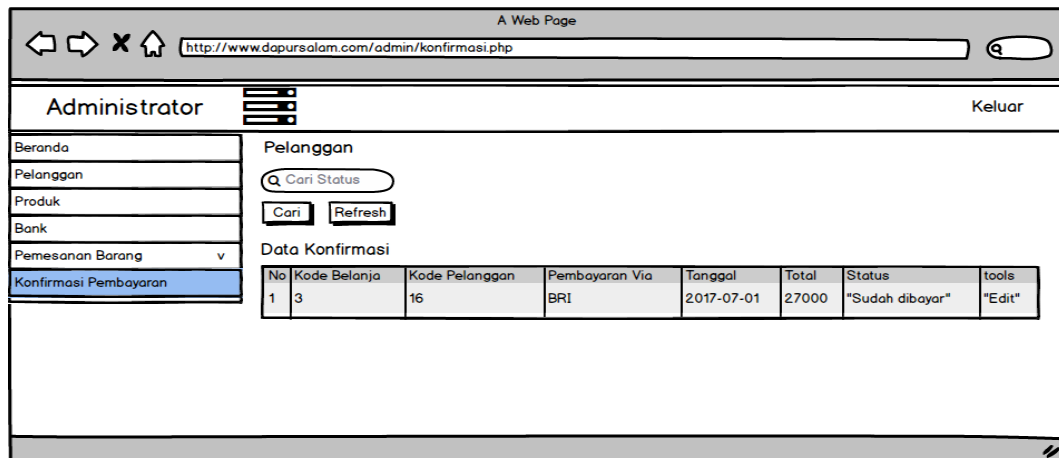
Rancangan antarmuka halaman edit order diterima administrator berfungsi untuk mengubah status pemesanan dari proses ke terkirim. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.35.



Gambar 3.35 Rancangan Halaman Edit Order Terkirim

### 3.6.26 Rancangan antarmuka halaman data konfirmasi administrator

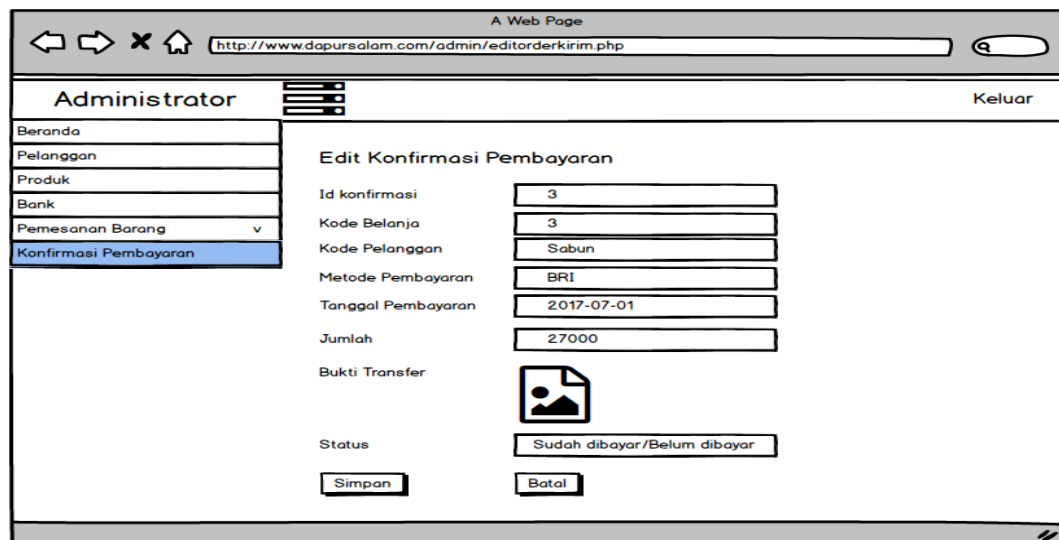
Rancangan antarmuka halaman data konfirmasi berfungsi untuk menampilkan data-data pesanan yang belum dibayar ataupun sudah dibayar sehingga dapat melanjutkan ke proses order dikirim. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.36.



Gambar 3.36 Rancangan Halaman Data Konfirmasi Administrator

### 3.6.27 Rancangan antarmuka halaman edit konfirmasi status bayar

Rancangan antarmuka halaman edit konfirmasi bayar berfungsi untuk mengubah status pembayaran apabila pelanggan sudah melakukan transaksi. Gambaran rancangan antarmuka halaman *website* dapat dilihat pada gambar 3.37.



Gambar 3.37 Rancangan Halaman Edit Konfirmasi Status Pembayaran