# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1. Perancangan Sistem

Perencanaan sistem adalah proses dimana sebuah sistem dirancang dan nantinya untuk dilakukan pengembangan sistem. Perencanaan sistem ini menyangkut estimasi dari kebutuhan-kebutuhan fisik, tenaga kerja dan dana yang dibutuhkan untuk mendukung pengembangan sistem serta untuk mendukung operasinya setelah diterapkan.

Perancangan sistem dapat diartikan sebagai :

1. Tahap setelah analisis dari siklus pengembangan sistem.
2. Pendefinisian atas kebutuhan-kebutuhan fungsional.
3. Persiapan untuk rancang bangun implementasi.
4. Menggambarkan sistem yang akan dibentuk, berupa penggambaran perencanaan, pembuatan sketsa, pengaturan dari beberapa elemen terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
5. Konfigurasi komponen *software* dan *hardware* sistem.

### 4.1.1 Tujuan Perancangan Sistem

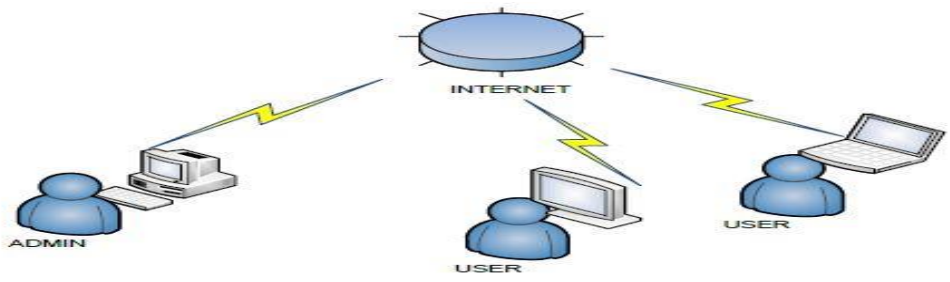
Perancangan sistem yang akan diaplikasikan ini bertujuan untuk memberikan gambaran secara umum kepada pengguna tentang sistem yang akan dibangun dan mengidentifikasikan komponen-komponen sistem

informasi yang akan didesain secara rinci. Tujuan perancangan sistem informasi penjualan online ini akan diuraikan sebagai berikut :

1. Sistem Informasi Penjualan Pada Toko AntzShop Berbasis Web, ditujukan untuk dapat memfasilitasi dan memudahkan pelanggan yang hendak membeli ataupun sekedar melihat-lihat produk AntzShop namun tidak dapat datang langsung ke toko.
2. Dengan adanya Sistem Informasi Penjualan Pada Toko AntzShop Berbasis *Web*, diharapkan dapat meningkatkan transaksi penjualan dan sekaligus sebagai media promosi yang lebih luas.

### 4.1.2 Gambaran Umum Sistem yang Diusulkan

Sistem yang diusulkan untuk pengolahan data penjualan produk Toko AntzShop dilakukan secara *online*, dimana para *user* dapat mengaksesnya langsung tanpa harus hadir ke lokasi, dalam hal ini Toko AntzShop. Sistem yang diusulkan pada *Web* ini bertindak sebagai *administrator* sedangkan *user* adalah para konsumen. Berikut adalah gambaran umum sistem yang diusulkan.



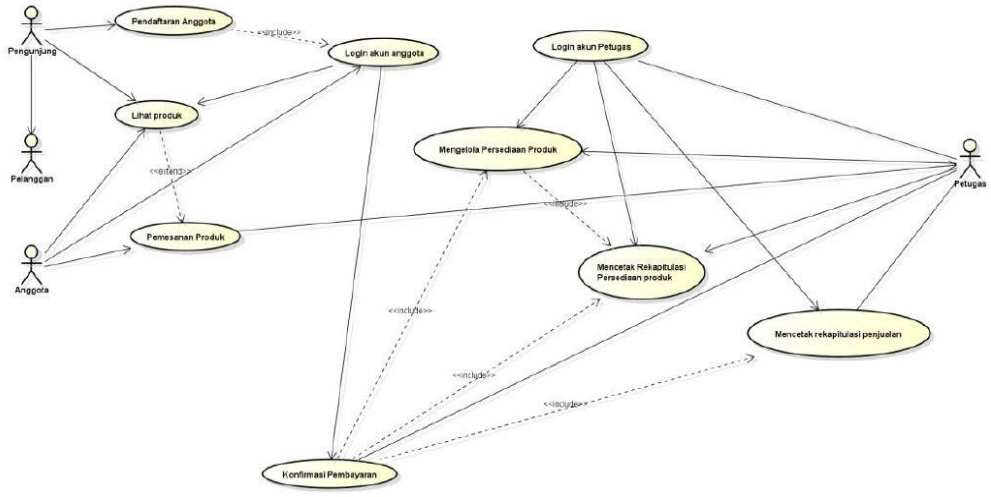

**Gambar 4.7** Gambaran Umum Sistem yang Diusulkan.

### 4.1.3 Perancangan Prosedur Yang Diusulkan

Perancangan ini mencakup *use case diagram*, *activity diagram*, *collaboration diagram*, *class diagram*, *object diagram*, *sequence diagram*, *component diagram dan deployment diagram* yang menghasilkan sistem lebih baik. Proses yang dirancang diuraikan menjadi beberapa bagian yang dapat membentuk sistem tersebut menjadi satu kesatuan komponen.

#### 4.1.3.1 *Uses Case*

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun, berikut adalah *use case* diagram dari sistem informasi penjualan yang diusulkan.




**Gambar 4.8.** *Uses Case* Diagram Sistem Informasi Penjualan Pada Toko AntzShop Berbasis *Web* Yang Diusulkan.

#### 4.1.3.2. Skenario *Uses case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. berikut skenario *use case* yang diusulkan.

1. Nama *Use Case* : Pendaftaran Anggota

   Aktor : Pengunjung

   Tujuan : Melakukan pendaftaran akun anggota

   Deskripsi : Dimana pengunjung melakukan proses pendaftaran yang kemudian data akunnya akan disimpan oleh sistem ke database

**Tabel 4.7.** Tabel Skenario *Use case* melakukan pendaftaran Anggota.

| Pengunjung | Sistem |
|---|---|
| 1. Masuk ke halaman pendaftaran akun anggota. | |
| 2. Melakukan pendaftaran. | |
| | 3. Menyimpan data akun anggota ke *database.* |

2. Nama *Use Case* : Lihat Produk

   Aktor : Pelangan (pengunjung,anggota)

   Tujuan : Melakukan Pendaftaran akun anggota

Deskripsi : Proses dimana pelangan melihat-lihat produk

**Tabel 4.8** Skenario *Use case* Lihat Produk.

| Pelanggan | Sistem |
|---|---|
| 1. Masuk ke halaman produk | |
| | 2. Menampilkan daftar produk yang tersedia di website |

3. Nama *Use Case* : *Login* akun anggota

   Aktor : Anggota

   Tujuan : Melakukan login akun anggota

   Deskripsi : Proses anggota melakukan login dengan mengisi *form login* dan akan di *verifikasi* oleh sistem

**Tabel 4.9** Skenario *Use case* Proses *Login* Anggota.

| Anggota | Sistem |
|---|---|
| 1. Masuk ke halaman *login* akun. | |
| 2. Melakukan *login* akun anggota dengan mengisi email dan kata sandi. | |
| | 3. Memverifikasi akun anggota. |
| | 4. Menampilkan halaman utama anggota. |

4. Nama *Use Case* : Pemesanan Produk

   Aktor : Anggota

Tujuan : Melakukan pemesanan produk

Deskripsi : proses dimana anggota melakukan pemesanan produk kemudian data pesanan tersebut akan di*verifikasi* oleh sistem

**Tabel 4.10** Skenario *Use case* Proses Pemesanan Produk.

| **Anggota** | **Sistem** |
|---|---|
| 1. Masuk ke halaman produk dan memilih produk yang akan di pesan | |
| 2. Mengisi form pemesanan | |
| | 3. Memverivikasi data pesanan. |
| | 4. Menyimpan data pesanan ke *database* |

5. Nama *Use Case* : Konfirmasi Pembayaran

   Aktor : Anggota

   Tujuan : Melakukan konfirmasi pembayaran berdasarkan pemesanan

   Deskripsi : Proses dimana anggota melakukan konfirmasi pembayaran setelah melakukan transfer.

**Tabel 4.11** Skenario *Use case* Proses Konfirmasi Pembayaran.

| Anggota | Sistem |
|---|---|
| 1. Masuk ke halaman konfirmasi pembayaran berdasarkan pemesanan. | |
| 2. Mengisi form konfirmasi. | |
| | 3. Memverivikasi data konfirmasi |
| | 4. Menyimpan data konfirmasi ke *database*. |

6. Nama *use case* : *Login* akun petugas

   Aktor : Petugas

   Tujuan : Melakukan *Login* akun petugas

   Deskripsi : Proses dimana petugas melakukan *login* terhadap sistem

**Tabel.4.12** Skenario *Use case* Proses *Login* Petugas

| **Petugas** | **Sistem** |
|---|---|
| 1. Masuk ke halaman *login* akun Petugas. | |
| 2. Melakukan *login* akun petugas dengan mengisi *usename* dan kata sandi. | |
| | 3. Memverifikasi akun petugas. |
| | 4. Menampilkan halaman utama petugas. |

7. Nama *use case* : Mengelola persedian produk

   Aktor : Petugas

   Tujuan : Memanipulasi Persedian Produk

   Deskripsi : Proses dimana dilakukannya perubahan data produk oleh petugas

**Tabel 4.13.** Skenario *Use case* Mengelola Persediaan Produk

| Petugas | Sistem |
|---|---|
| 1. Masuk ke halaman produk. | |
| 2. Memanipulasi data-data atau informasi produk. | |
| | 3. Menyimpan perubahan data produk ke *database*. |

8. Nama *use case* : Mencetak rekapitulasi persedian produk

   Aktor : Petugas

   Tujuan : Membuat dan mencetak rekapitulasi persediaan produk

   Deskripsi : Proses dimana petugas membuat laporan persedian produk dan mencetaknya.

**Tabel 4.14.** Skenario *Use case* Mencetak Rekapitulasi Persedian Produk.

| Petugas | Sistem |
|---|---|
| 1. Masuk ke halaman produk dan membuat rekapitulasi persediaan produk. | |

| | |
|---|---|
| | 2. Menampilkan data-data produk. |
| | 3. Mencetak rekapitulasi persediaan Produk. |

9. Nama *use case* : Mencetak rekapitulasi transaksi penjualan produk

   Aktor : Petugas

   Tujuan : Membuat dan mencetak rekapitulasi transaksi penjualan produk

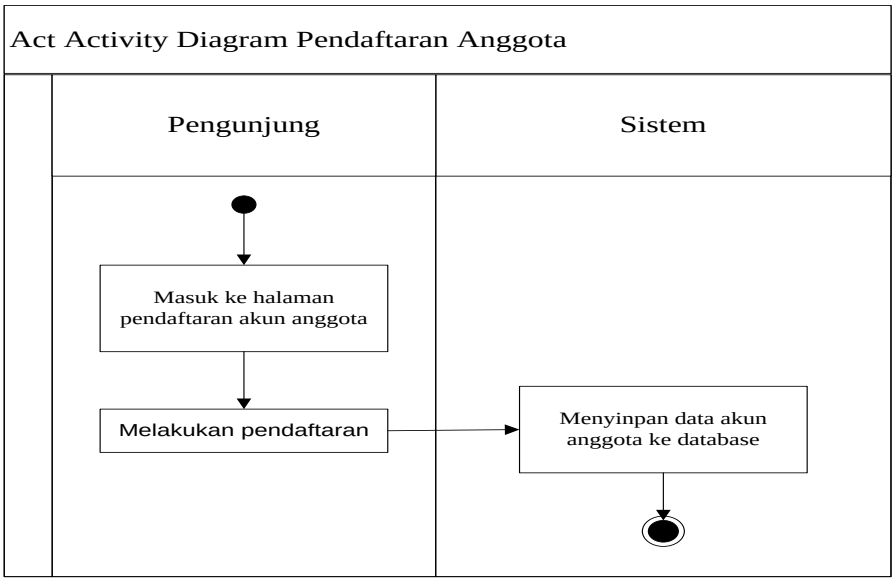
   Deskripsi : Proses dimana petugas membuat rekapitulasi transaksi penjualan produk dan mencetaknya

**Tabel 4.15.** Skenario *Use case* Mencetak rekapitulasi transaksi penjualan

| **Petugas** | **Sistem** |
|---|---|
| 1. Masuk ke halaman transaksi penjualan dan membuat rekapitulasi transaksi penjualan. | |
| | 2. Menampilkan data-data transaksi penjualan. |
| | 3. Mencetak rekapitulasi transaksi penjualan. |

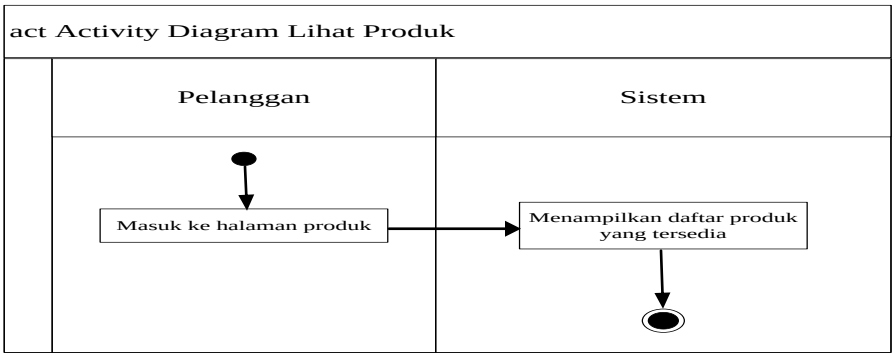
#### 4.1.3.3 *Activity Diagram*

*Activity diagram* berikut ini memperlihatkan secara rinci aliran data secara logika tanpa mempertimbangkan lingkungan fisik dimana data mengalir. Berikut *activity diagram* yang diusulkan pada toko AntzShop.



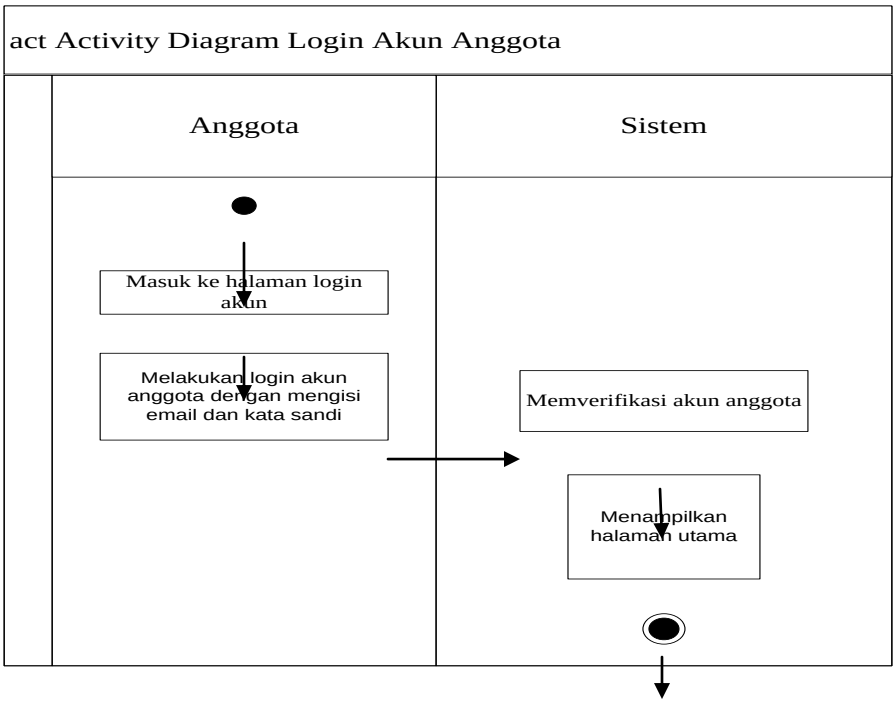

**Gambar 4.9.** *Activity Diagram* Pendaftaran anggota pada toko AntzShop Berbasis *Web* yang Diusulkan.

Pada *activity diagram* pendaftaran anggota ini, dimana pegunjung masuk ke halaman pendaftaran akun anggota dan melakukan pendaftaran dengan cara mengisi form pendaftaran, lalu sistem menyimpan data akun anggota ke *database*.



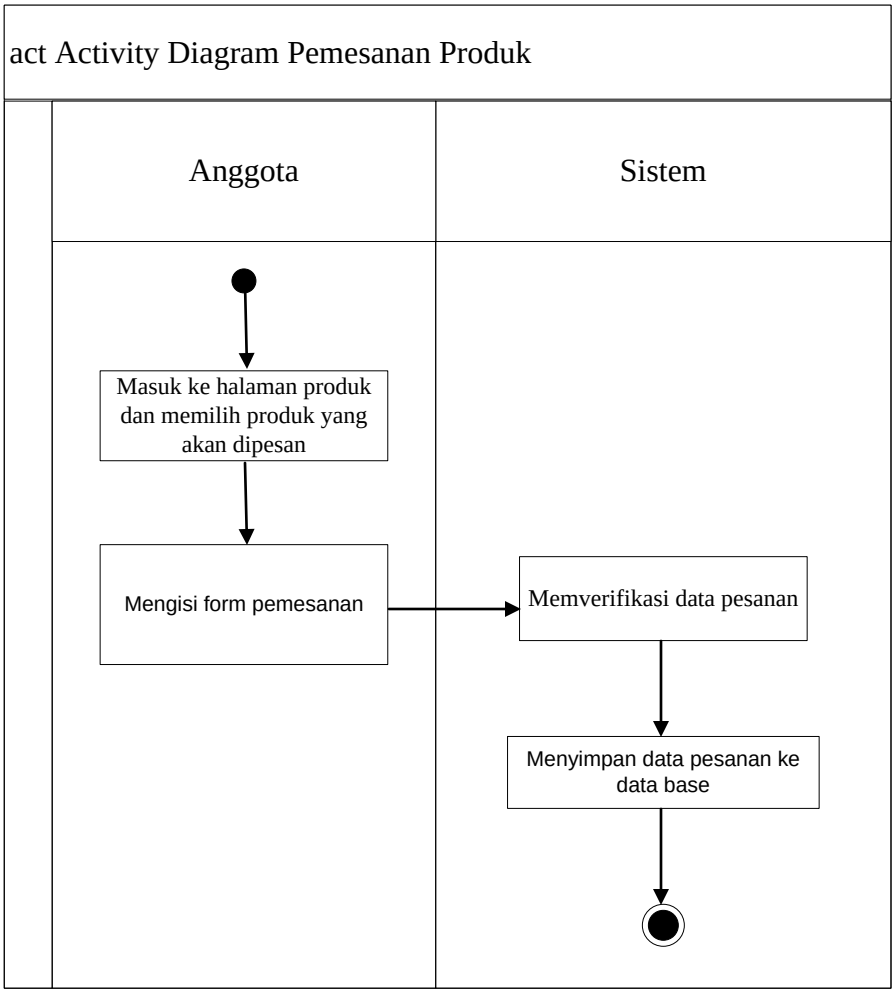

**Gambar 4.10.** *Activity Diagram* lihat Produk pada toko AntzShop Berbasis *Web* yang Diusulkan.

Dalam diagram ini, menggambarkan dimana pelanggan masuk ke halaman produk dan sistem secara otomatis menampilkan daftar produk yang tersedia.



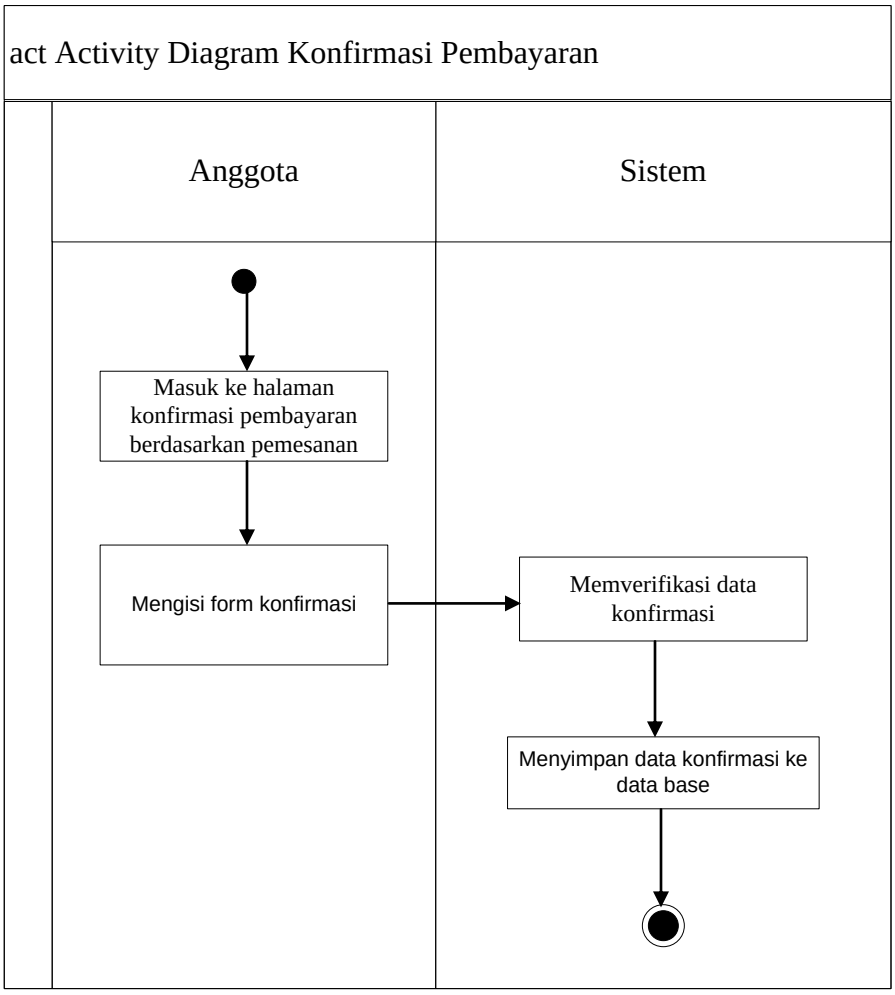

**Gambar 4.11.** *Activity Diagram* Login akun anggota pada toko AntzShop Berbasis Web yang Diusulkan.

Pada *activity diagram login* akun anggota ini, menggambarakan bagaimana anggota masuk ke halaman *login* anggota dan melakukan *login* anggota dengan mengisi *email* dan kata sandi, kemudian sistem memverifikasi akun anggota tersebut dan menampilkan halaman utama anggota.




**Gambar 4.12.** *Activity Diagram* Pemesanan Produk pada toko AntzShop Berbasis *Web* yang Diusulkan.

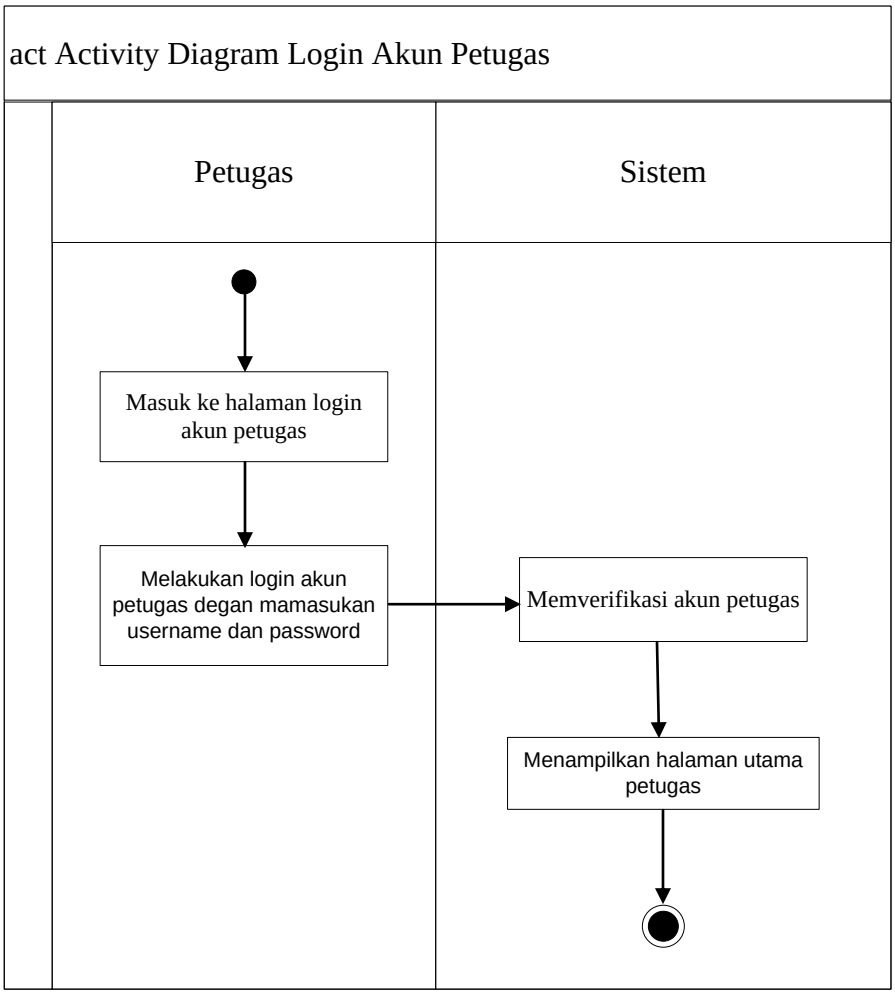
Pada *activity diagram* ini, menggambarkan dimana anggota masuk kehalaman produk dan memilih produk yang akan dipesan lalu mengisi form pemesanan. Kemudian sistem memverifikasi data pesanan, kemudian menyimpan data pesanan tersebut ke *database*.




**Gambar 4.13.** *Activity Diagram* Konfirmasi Pembayaran pada toko AntzShop Berbasis *Web* yang Diusulkan.

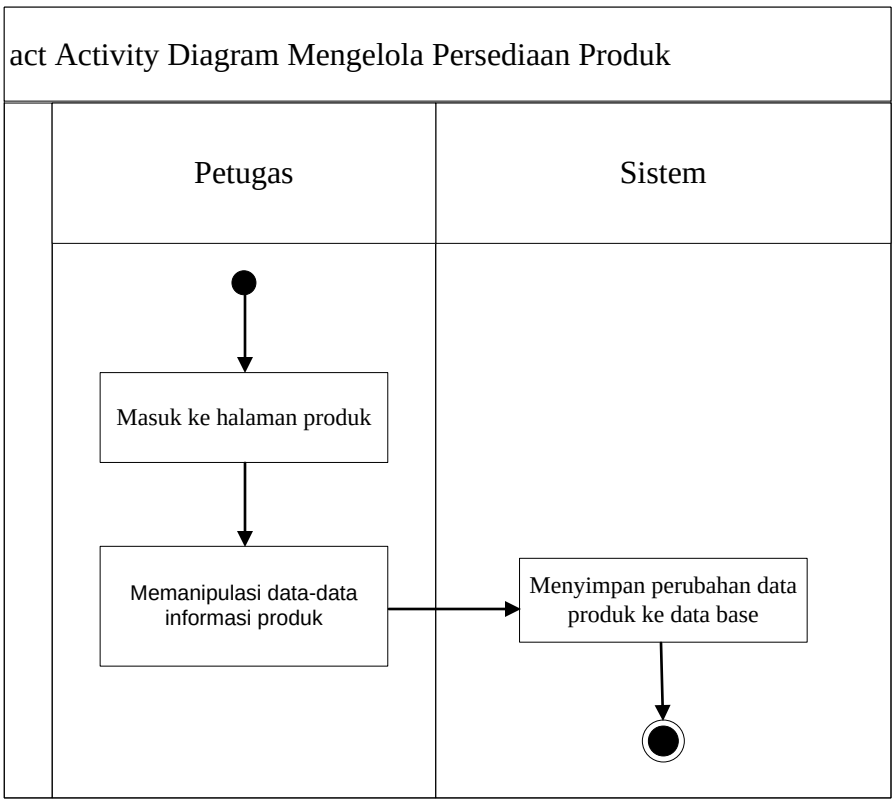
Anggota masuk kehalaman konfirmasi pembayaran berdasarkan pemesanan, kemudian mengisi form konfirmasi dan

sistem memverifikasi data konfirmasi kemudian menyimpan data konfirmasi tersebut ke *database*.



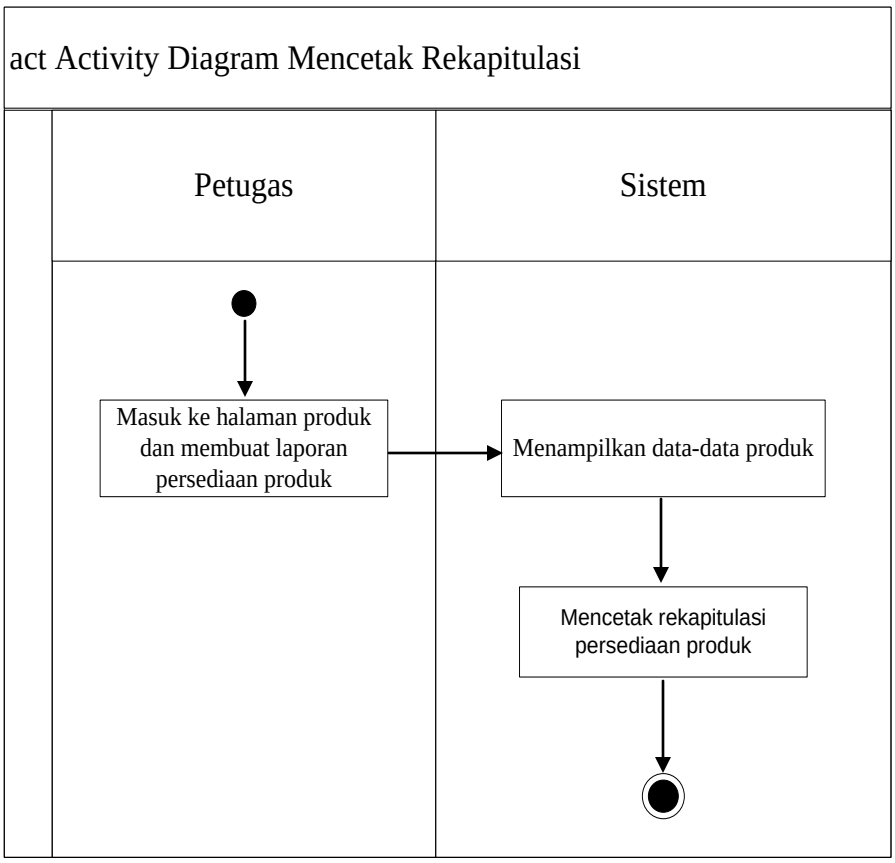

**Gambar 4.14.** *Activity Diagram Login* Petugas pada toko AntzShop Berbasis *Web* yang Diusulkan.

Petugas masuk ke halaman *login* petugas dan melakukan *login* akun petugas dengan mengisi *username* dan kata sandi, kemudian sistem memverifikasi akun petugas tersebut lalu menampilkan halaman petugas.



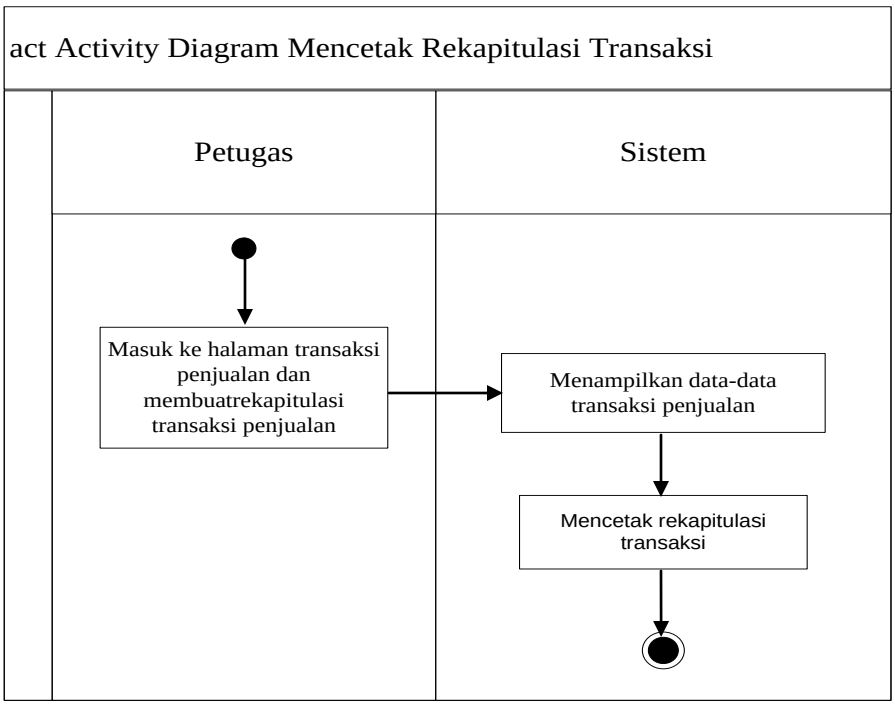

**Gambar 4.15.** *Activity Diagram* Mengelola persediaan produk pada toko AntzShop Berbasis Web yang Diusulkan.

Petugas masuk ke halaman produk dan memanipulasi data-data/ informasi produk, kemudian sistem menyimpan perubahan data produk tersebut ke *database.*




**Gambar 4.16.** *Activity Diagram* Mencetak rekapitulasi persediaan produk pada toko AntzShop Berbasis *Web* yang Diusulkan.

Petugas disini masuk ke halaman produk dan membuat laporan persediaan produk, kemudian sistem menampilkan data-data produk dan mencetak laporan persediaan produk.




**Gambar 4.17.** *Activity Diagram* Mencetak Rekapitulasi Transaksi Penjualan pada Toko AntzShop Berbasis *Web* yang Diusulkan.

Petugas masuk ke halaman transaksi penjualan dan membuat laporan penjualan, kemudian sistem menampilkan data-data transaksi penjualan dan mencetak laporan transaksi.

#### 4.1.3.5 *Sequence Diagram*

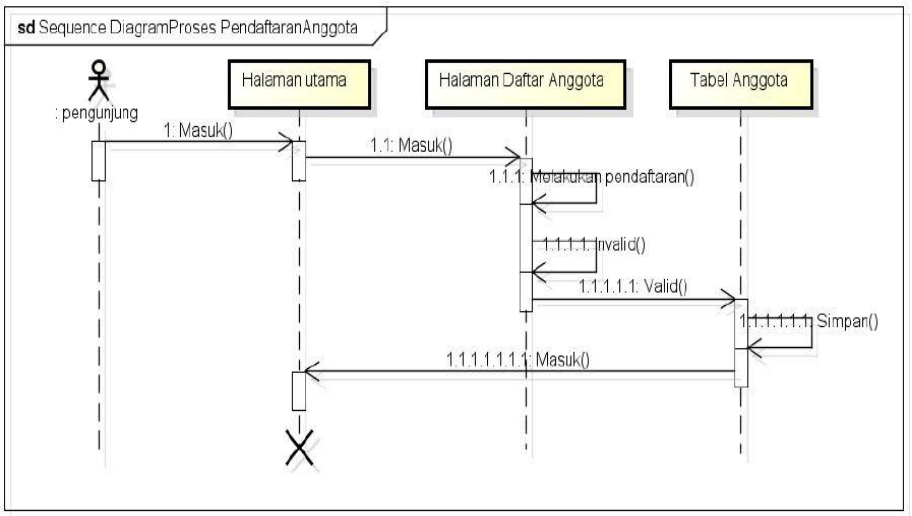
*Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya sequence Diagram selain digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada

titik tertentu dalam eksekusi sistem. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan proses vertikal.

Berikut adalah *sequence diagram* yang ada pada sistem penjualan pada toko AntzShop, yaitu:

**1. *Sequence Diagram* Pendaftaran Member**

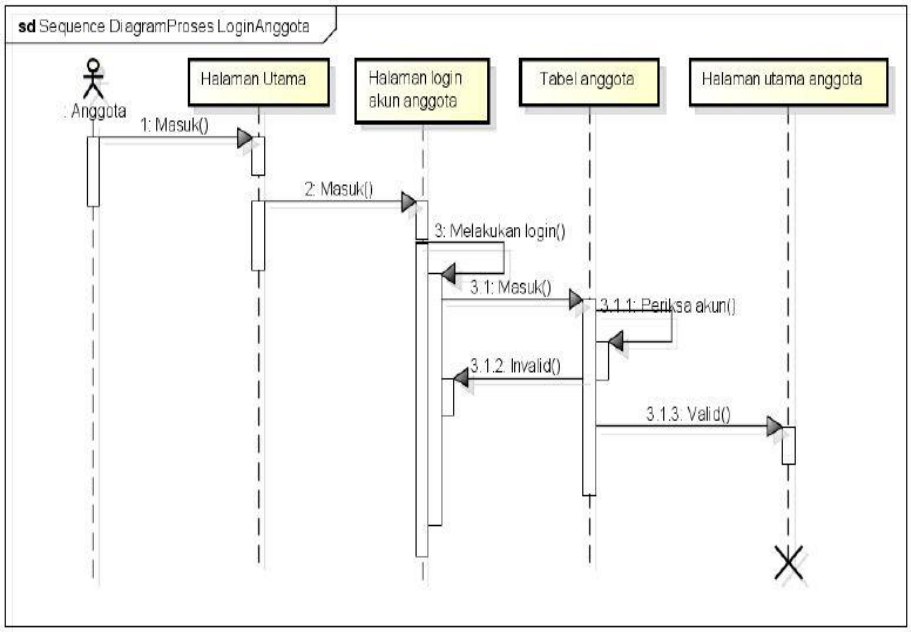
Pengunjung masuk ke halaman utama web AntzShop, apabila pengunjung ingin dapat melakukan transaksi maka harus mendaftar terlebih dahulu. Dengan cara masuk ke halaman daftar anggota dan melakukan pendaftaran dengan mengisikan data di form pendaftaran. Kemudian sistem akan memvalidasi data tersebut dan menyimpanya ke *database*.




**Gambar 4.18.** *Sequence Diagram* Proses Pendaftran Anggota yang diusulkan.

**2. *Sequence Diagram* Untuk Login Anggota**

Anggota masuk kehalaman utama, kemudian masuk ke halaman *login* akun anggota untuk melakukan proses *login*. Kemudian sistem akan memvalidasi akun *login* tersebut, sehingga jika valid anggota dapat masuk ke halaman utama anggota untuk melakukan proses selanjutnya.

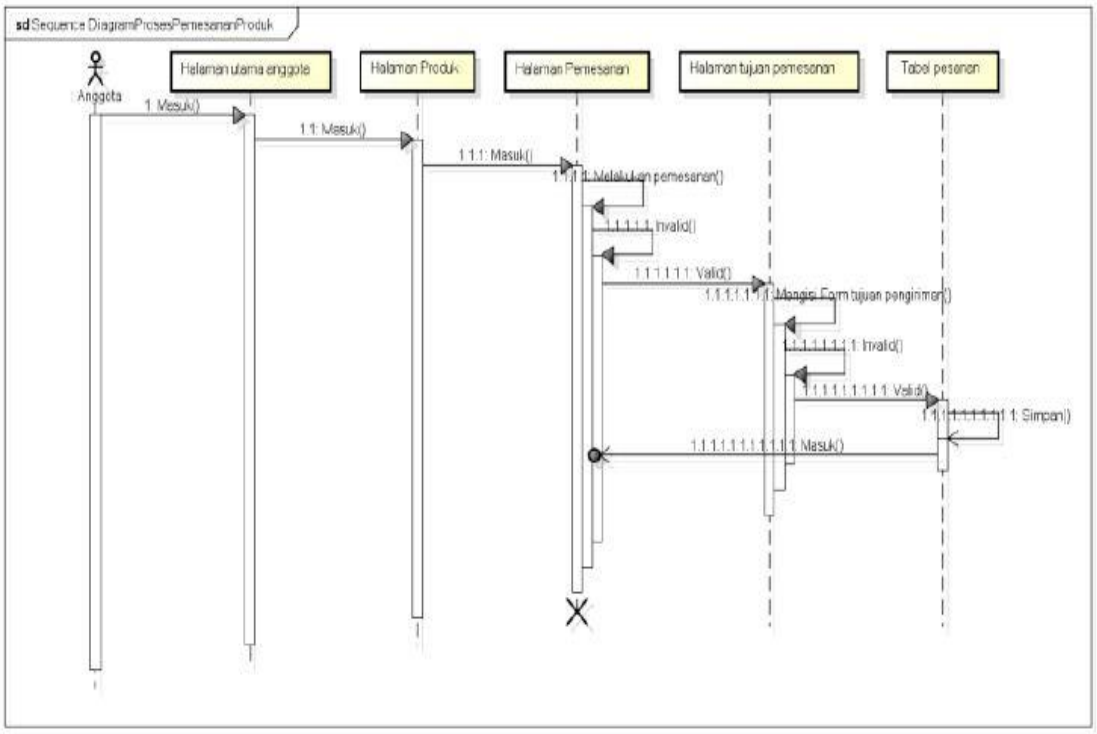



**Gambar 4.19.** *Sequence Diagram* Proses Login Anggota yang diusulkan.

**3. *Sequence Diagram* Untuk Pemesanan Produk**

Pada proses ini dimana anggota masuk ke halaman utama anggota setelah melakukan proses login terlebih dahulu, kemudian masuk ke halaman produk dan melakukan pemesanan dihalaman

pemesanan. Kemudian sistem akan memvalidasi pemesanan tersebut, jika valid maka akan langsung diarahkan kehalaman tujuan pemesanan untuk mengisi form tujuan pengiriman, lalu sistem melakukan validasi untuk tujuan pemesanan.jika valid maka data pemesanan akan tersimpan kedalam table pemesanan.

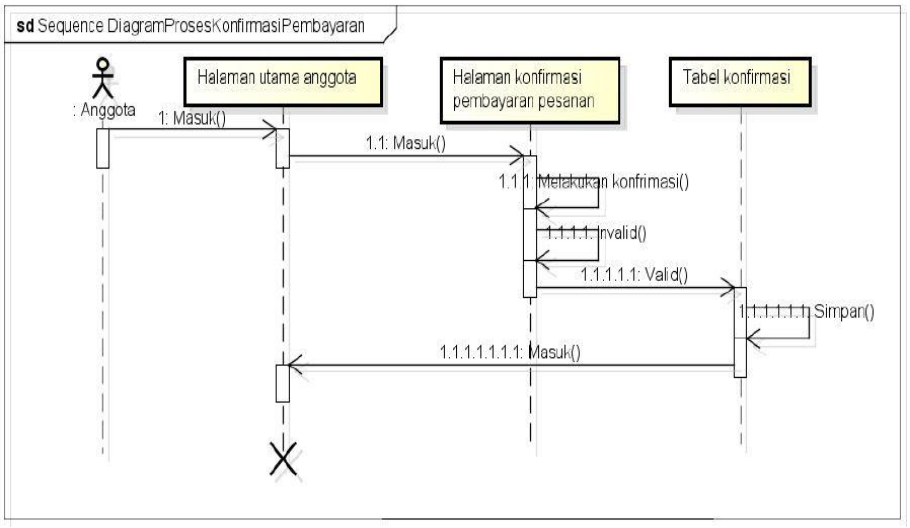



**Gambar 4.20.** *Sequence Diagram* Proses Pemesanan Produk yang diusulkan.

**4. *Sequence Diagram* Untuk Konfirmasi Pembayaran oleh Anggota**

Pada diagram ini menggambarkan bagaimana proses konfirmasi pembayaran yang dilakukan oleh anggota dimana anggota terlebih dahulu masuk kehalaman utama anggota, kemudian masuk kehalaman konfirmasi pembayaran pemesanan, dan mengisi form

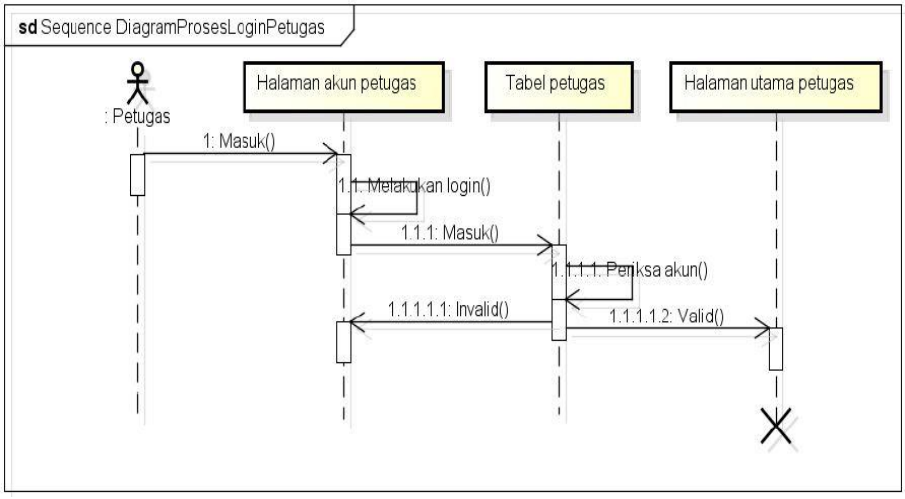
konfirmasi pemesanan. Kemudian sistem melakukan validasi, jika data yang di isi valid maka data tersebut akan tersimpan kedalam tabel konfirmasi.




**Gambar 4.21.** *Sequence Diagram* Proses Konfirmasi Pembayaran oleh Anggota yang diusulkan.

### 5. *Sequence Diagram* Untuk Login Petugas

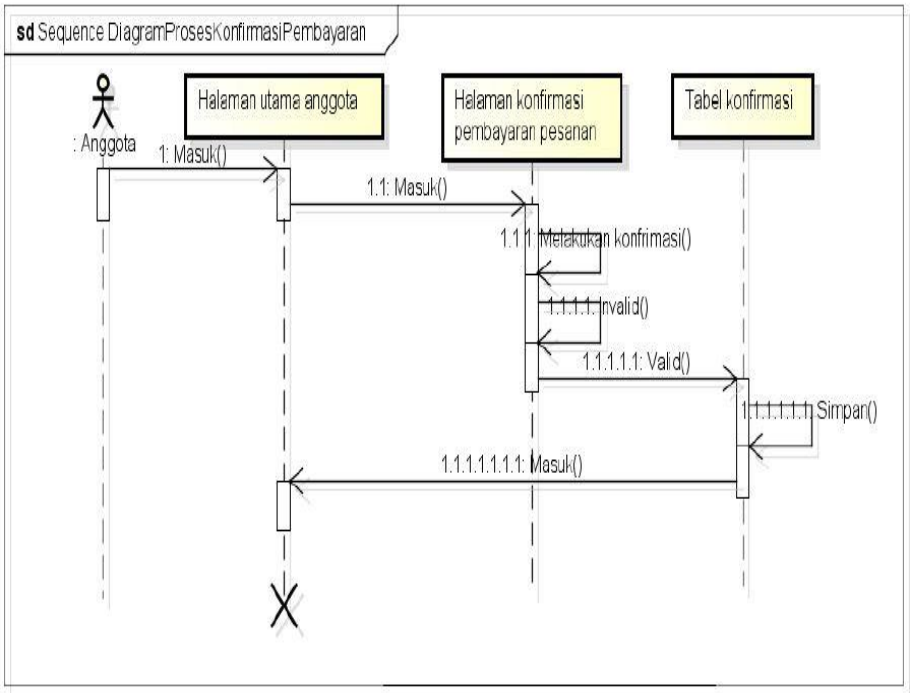
Dimana petugas masuk ke halaman *login* akun petugas untuk melakukan proses *login*. Kemudian sistem akan memvalidasi akun *login* tersebut, sehingga jika valid petugas dapat masuk ke halaman utama petugas untuk melakukan proses selanjutnya.




**Gambar 4.22.** *Sequence Diagram*Proses *Login* Petugas yang Diusulkan.

### 6. *Sequence Diagram* Untuk Konfirmasi Pembayaran oleh Petugas

Petugas masuk kehalam utama petugas, kemudian masuk kehalaman konfirmasi hingga masuk ke halaman *acc* konfirmasi untuk melakukan konfirmasi pembayaran yang telah dilakukan oleh anggota. kemudian data yang tersimpan ke dalam tabel konfirmasi dan sistem langsung menampilkan halaman tabel konfirmasi.

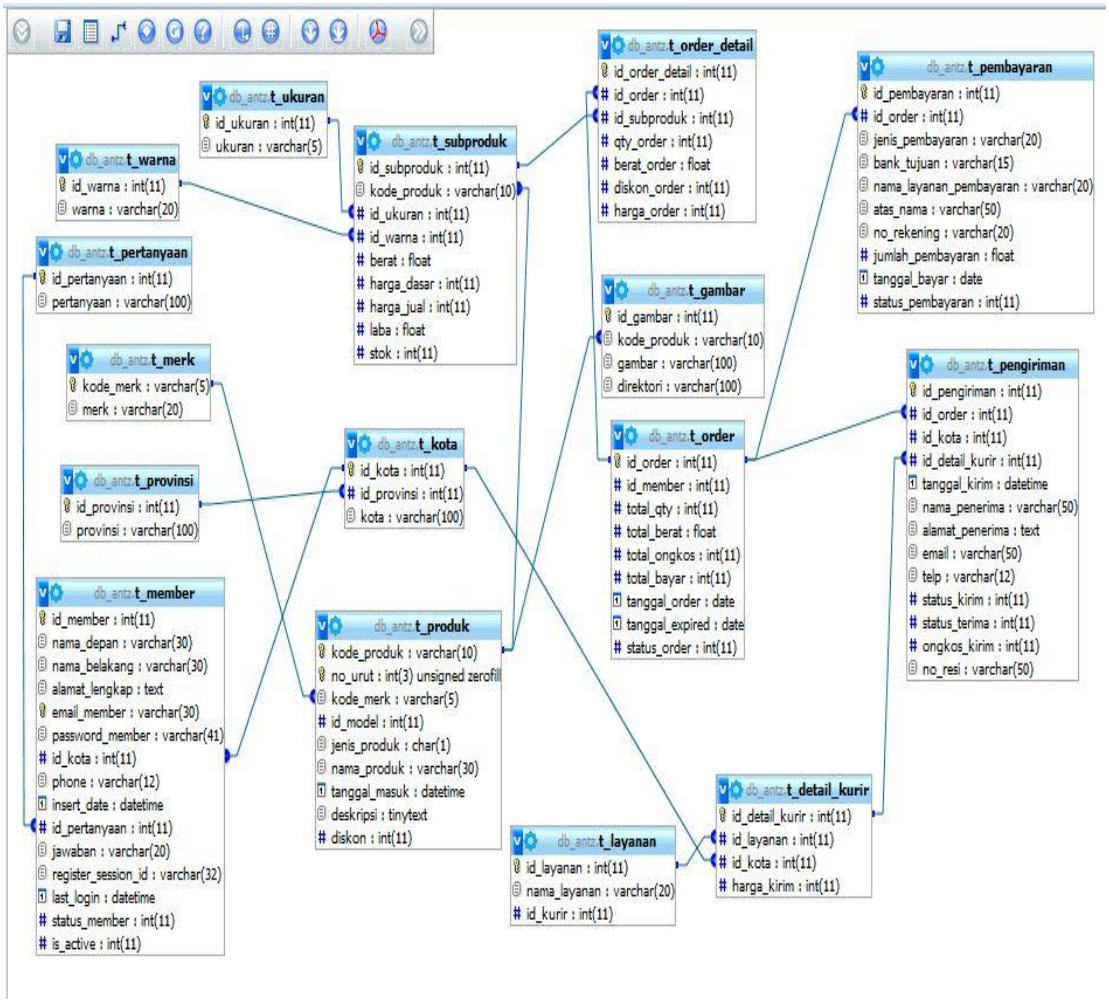



**Gambar 4.23.** *Sequence Diagram* Proses Konfirmasi Pembayaran oleh Petugas yang Diusulkan.

### 4.1.3.5 *Class* Diagram

*Class diagram* bertujuan untuk menggambarkan struktur statis *class* di dalam sistem. *Class* berfungsi untuk merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem pemasaran produk mulai dari bagian pendaftaran sampai dengan pemesanan beserta proses-proses yang terjadi, maka dapat dibuat C*lass Diagram* Berikut *Class Diagram* Sistem Informasi Penjualan pada toko AntzShop

.

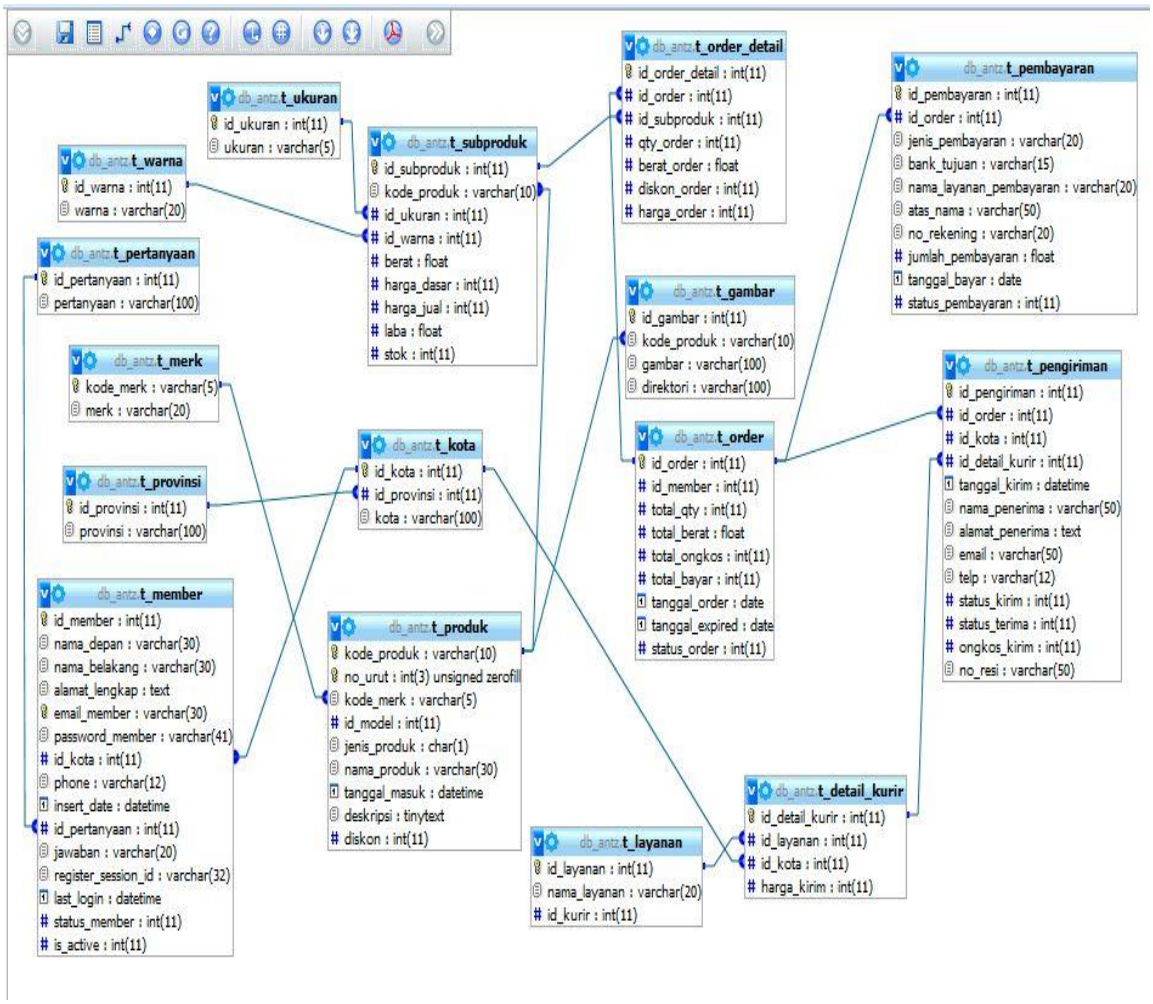



**Gambar 4.24.** *Class Diagram* Sistem Informasi Penjualan pada toko AntzShop yang Diusulkan.

#### 4.1.3.6 *Object* Diagram

Diagram objek ialah merupakan suatu diagram yang mengambarkan objek-objek dalam suatu sistem. Bila pada kelas diagram terdapat banyak relasi antar kelas, namun pada

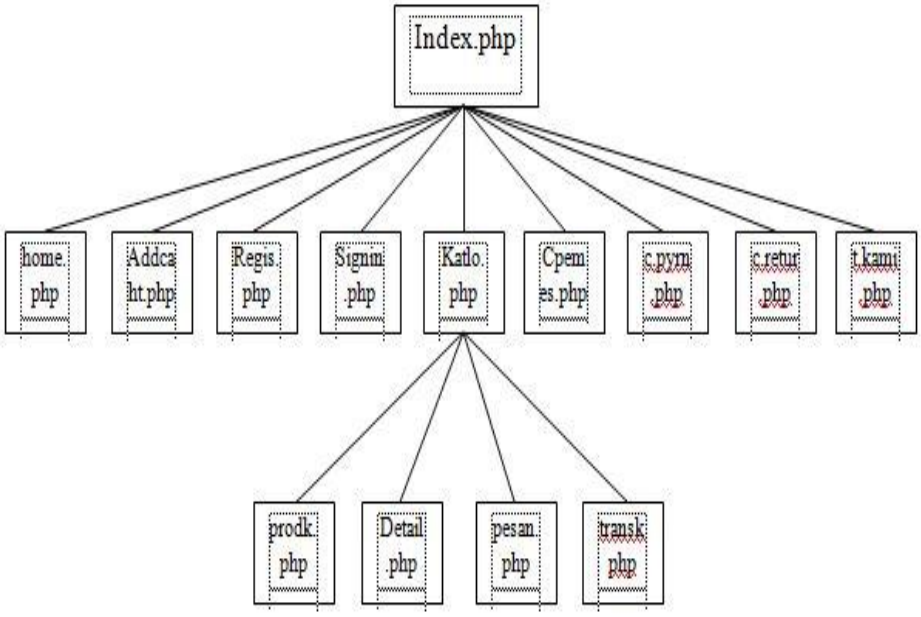
diagram objek, hanya terdapat satu relasi antar kelas, yaitu asosiasi. Tidak hanya itu, pada diagram objek yang didefinisikan hanya nama objek dalam suatu kelas beserta atributnya saja, tanpa operasi yang dilakukan oleh objek tersebut.




**Gambar 4.25.** *Object Diagram* Sistem Informasi Penjualan pada toko AntzShop yang Diusulkan.

#### 4.1.3.7 *Component* Diagram

*Component* adalah unit fisik yang nyata yang menjadi bagian dari *deployment independent*. *Component* ini diimplementasikan meskipun pada sistem yang kecil. Perancangan menu pada aplikasi yang diusulkan digambarkan oleh gambar *component diagram* berikut.

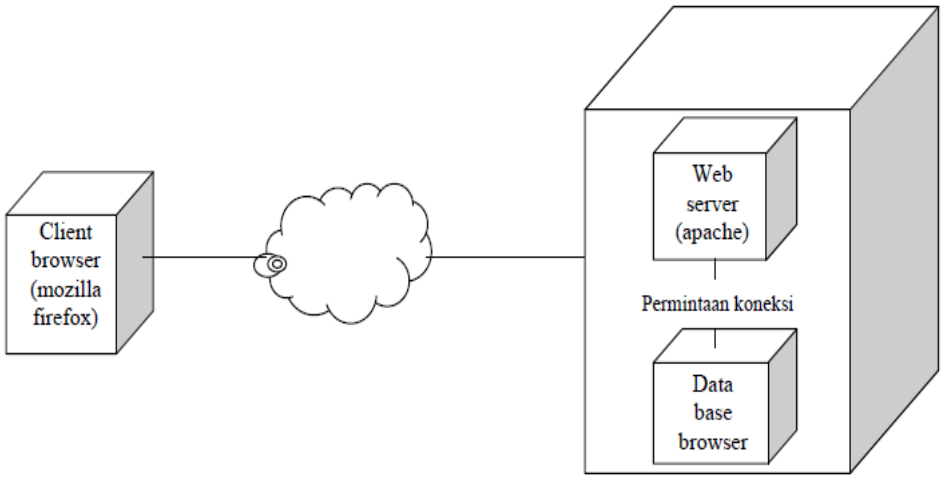



**Gambar 4.26.** *Component Diagram* Sistem Informasi Penjualan pada toko AntzShop yang Diusulkan.

#### 4.1.3.8 *Deployment* Diagram

Asumsi yang digunakan pada perangkat lunak ini memakai sistem yang *stand alone*, artinya tidak ada satu

komponenpun yang akan dibagi pakai *(sharing)* dengan aplikasi *web* yang lain, oleh karena itu tidak perlu *application server* terpisah




**Gambar 4.27.** *Deployment Diagram* Sistem Informasi Penjualan pada toko AntzShop yang Diusulkan.

## 4.2. Perancangan Antar Muka

Perancangan ini merupakan desain utama sebagai *user interface* yang akan digunakan oleh sistem yang akan dirancang.

<table>
<tr><td colspan="2">Login akun</td></tr>
<tr><td>Banner</td><td rowspan="2">Menu</td></tr>
<tr><td>Informasi</td></tr>
<tr><td colspan="2">Footer</td></tr>
</table>

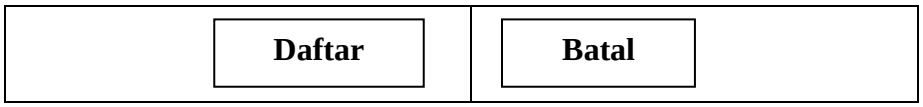
**Gambar 4.28.** Perancangan antar muka .

### 4.2.1 Perancangan Input

Perancangan input dirancang sebagai media dimana pengguna atau administrator dapat dengan mudah menyimpan data yang nantinya akan diperlukan baik itu oleh sistem maupun oleh pengguna itu sendiri. Perancangan input dari perangkat lunak yang akan dibangun adalah sebagai berikut :

**1. Pendaftaran Anggota**

| | |
|---|---|
| Nama Depan : | |
| Nama Belakang ; | |
| Alamat : | |
| Provinsi : | v |
| Kota : | v |
| Phone : | |
| **Data Login** | |
| Email : | |
| Sandi : | |
| Ulangi sandi : | |
| **Pertanyaan keamanan** | |
| Pertanyaan : | v |
| Jawaban : | |
| Pesyaratan : | |




**Gambar 4.29.** Perancangan Form *Input* Pendaftran Anggota

**2. Login Anggota**

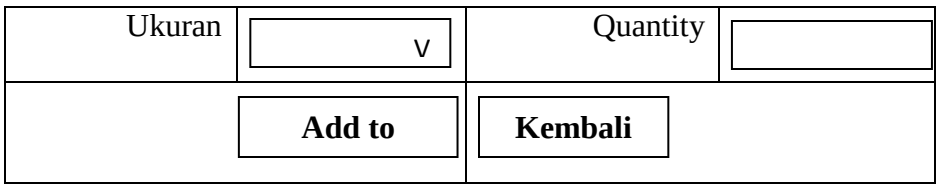
Perancangan form ini sebagai tahap selanjutnya dari pengguna yang telah menjadi anggota untuk dapat melanjutkan ke tahap selanjutnya dengan melakukan *login* terlebih dahulu dengan mengisikan *username* dan *password*.




**Gambar 4.30.** Perancangan Tampilan Halaman *Login* Anggota.

**3. Pemesanan Produk**

Berikut merupakan perancangan form dimana anggota dapat melakukan proses pemesanan dengan memilih produk yang akan dipesan terlebih dahulu dan menentukan detail dari produk dan pesanan tersebut.




**Gambar 4.31.** Perancangan Tampilan Halaman Pemesanan Produk.

**4. Konfirmasi Pembayaran**

Perancangan form ini bertujuan sebagai konfirmasi pembayaran dari anggota yang telah melakukan pembayaran atas produk yang telah dipesannya.

| | |
|---|---|
| ID order | |
| Total pembayaran | |
| tanggal pembayaran | |
| Bank tujuan | V |
| BankAsal | |
| No.Rekening | |
| Atas nama | |
| | **Kirim** |

**Gambar 4.32.** Perancanagan Tampilan Konfirmasi Pembayaran Pesanan.

**5. Form Login Petugas**

Sama halnya dengan perancangan form *login* anggota, form *login* petugas inipun bertujuan untuk memberikan hak akses ke pada petugas untuk melakukan pengolahan data dengan menginputkan *username* dan *password*.

| | | |
|---|---|---|
| **Username** | **Password** | **Login** |

**Gambar 4.33.** Perancangan Tampilan Form *Login* Petugas.

**6. Form Tambah Produk**

Perancangan form input tambah produk ini bertujuan agar petugas dapat menambah jumlah dari suatu produk yang akan ditawarkan oleh AntzShop

| | |
|---|---|
| Kategori | v |
| Model | v |
| Merk | v |
| Jenis produk | v |
| Nama produk | |
| Diskon | |
| Deskripsi | |
| **Simpan** | |

**Gambar 4.34.** Perancangan Tampilan Form Tambah Produk.

### 4.2.3. Perancangan Output

Perancangan *output* merupakan rencana pembuatan antarmuka yang bertujuan untuk menampilkan keluaran program. Berikut ini adalah perancangan *output* pada sistem yang akan dirancang.

**1. Tampilan Produk**

Perancangan *output* ini bertujuan untuk menampilkan daftar dari produk-produk baik yang ditawarkan kepada pelanggan atau pun sebagai *output* saja bagi petugas untuk melakukan manipulasi terhadap data tersebut.

| | |
|---|---|
| Gambar | Nama Produk |
| | Harga |
| | Detail |

**Gambar 4.35.** Perancangan Tampilan Produk.

**2. Tampilan Detail Produk**

Perancangan *output* ini bertujuan untuk menampilkan daftar detail produk baik yang ditawarkan kepada pelanggan atau pun sebagai *output* saja bagi petugas untuk melakukan manipulasi terhadap data tersebut.

| Gambar | Ukuran | Stok | Berat(kg) | Diskon(%) | Harga |
|---|---|---|---|---|---|
| | xxxx | xx | xx | xx | xxxx |

**Gambar 4.36.** Perancanagan Tampilan Detail Produk.

**3. Tampilan Daftar Keranjang**

Perancangan tampilan ini bertujuan untuk menampilkan semua daftar pembelanjaan yang belum di konfirmasi kan oleh anggota.

| Produk | Hapus | Ukuran | Berat (kg) | Qty | Diskon(%) | Harga | Sub Total |
|---|---|---|---|---|---|---|---|
| Jasa pengiriman yang akan digunakan | | | | | | | |

| Pilih jasa pengiriman | Pilih layanan cargo | Pilih kota tujuan |
|---|---|---|
| V | V | V |
| **Belanja lagi** **Lanjutkan** | | |

**Gambar 4.37** Perancangan Tampilan Daftar Keranjang.

**4. Tampilan Daftar Pesanan Produk**

Perancangan daftar pesanan produk ini bertujuan untuk menampilkan semua daftar pesanan yang telah dilakukan oleh anggota sebelum dilanjutkan ke tahap selanjutnya.

| ID order | Tanggal pemesanan | Tanggal expired | Status pesanan | Jumlah bayar | Aksi |
|---|---|---|---|---|---|
| xx | xxxx | xxxx | xxxx | xxxx | xxxx |

**Gambar 4.38.** Perancangan Tampilan Daftar Pesanan Produk.

**5. Tampilan Daftar Konfirmasi Pemesanan**

Perancangan tampilan ini bertujuan untuk menampilkan semua daftar pesanan baik yang sudah maupun yang belum dikonfirmasi oleh petugas.

| No orderan | Tanggal pesanan | Tanggal expired | Member | Status pesanan | Total bayar | Aksi |
|---|---|---|---|---|---|---|
| xx | xxxx | xxxx | xx | xx | xxxx | xx |

**Gambar 4.39** Perancangan Tampilan Daftar Konfirmasi Pesanan

**6. Tampilan Perubahan Status Kelola Pembayaran**

Perancangan tampilan ini bertujuan untuk menampilkan perubahan status kelola pembayaran yang sudah melakukan pembayaran, pada form ini dilakukan oleh petugas.

| Data Konfirmasi Pembayaran | | |
|---|---|---|
| Tanggal Bayar | : | |
| Jenis Pembayaran | : | |
| Bank Tujuan | : | |
| Bank Asal | : | |
| No.Rekening/transfer/IPN Paypal | : | |
| Atas Nama | : | |
| Jumlah Bayar | : | |
| **Sudah Bayar V** | | |

**Gambar 4.40** Perancangan Tampilan Perubahan Satus Kelola Pembayaran.

**7. Tampilan Input Nomer Resi**

Perancangan tampilan ini bertujuan untuk menginputkan nomer resi, menginformasikan kepada konsumen melalui email bahwa barang sudah dikirim.

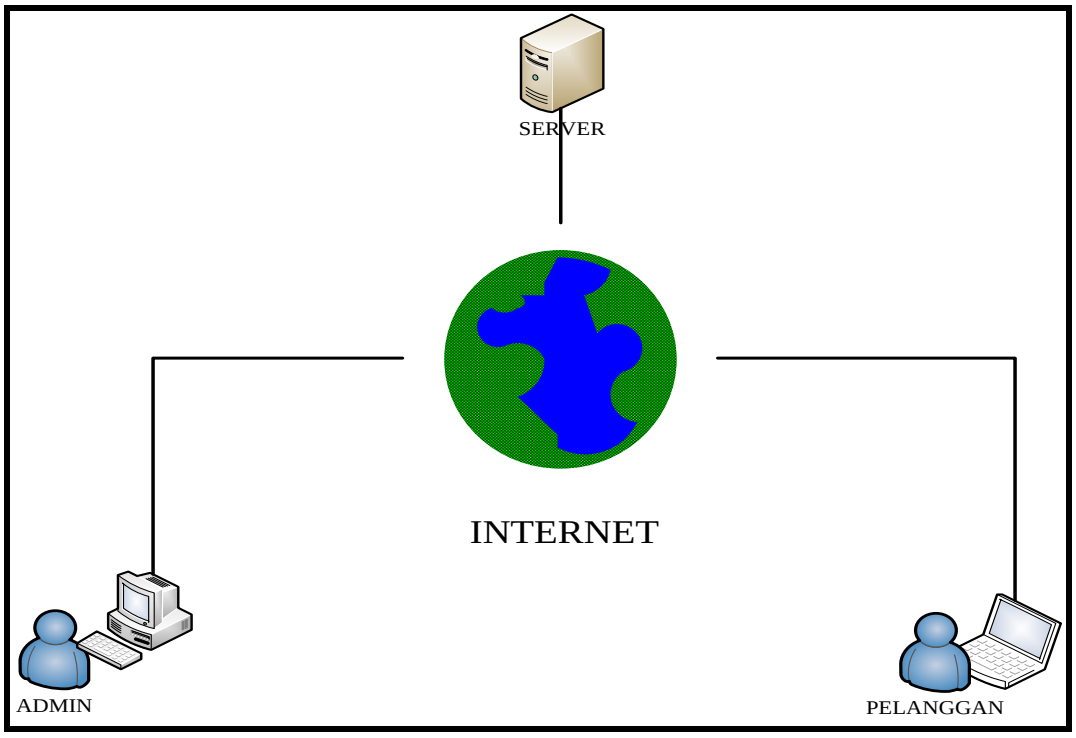
| Nomor Resi Pengiriman | | |
|---|---|---|
| Nomor Resi | : | |
| | | **Kirim** |

**Gambar 4.41** Perancangan Tampilan Perubahan Satus Kelola Penjualan

## 4.3. Perancangan arsitektur jaringan

Sistem informasi penjualan ini digunakan oleh beberapa pengguna dengan hak akses tersediri, sehingga untuk mengakses sistem informasi penjualan ini diperlukan

koneksi internet, koneksi internet ini sangatlah dbutuhkan karena sistem informasi ini sendiri merupakan sistem informasi berbasis website yang diharuskan online dalam penggunaanya baik dari sisi admin maupun pelanggan. Dengan menggunakan bahasa pemrograman PHP dan MYSQL sebagai databasenya maka sistem informasi ini sendiri akan terhubung ke sisi server. Adapun gambaran arsitektur jaringannya adalah sebagai berikut :




**Gambar 4.29** Rancangan Arsitektur Jaringan

## 4.4. Implementasi

Implementasi *prototype* sistem informasi AntzShop dilakukan menggunakan bahasa dan lingkungan pemrograman PHP dengan adobe dreamweaver 8 sebagai

perangkat lunaknya, dan juga bahasa SQL sebagai basis data dengan MySQL sebagai perangkat lunaknya. Bahasa pemograman PHP tersebut dapat dijalankan pada berbagai platform sistem operasi dan perangkat keras, tetapi implementasi dan pengujian sepenuhnya hanya dilakukan di perangkat keras *PC (personal computer)* dengan sistem operasi Microsoft Windows 7.

### 4.4.1 Batasan Inplementasi

Dalam mengimplementasikan perangkat lunak, batasan sebagai pengendalian aplikasi ini ada beberapa hal yang perlu dibatasai dalam implementasi ini yaitu :

1. Perangkat lunak yang digunakan dalam pengimplementasian basis data ini adalah MySQL yang telah terkompilasi kedalam sebuah perangkat lunak bebas yaitu xampp.
2. Tampilan untuk *web based* menggunakan bahasa Indonesia dan Inggris.

### 4.4.2 Implementasi Perangkat Lunak

Untuk implementasi perangkat lunak sitem informasi Penjualan pada Toko AntzShop ini digunakan xampp-win32-1.7.0-installer, Dreamweaver 8 dan MySQL (5.0.67) dan apache web *server*. Xampp dipilih sebagai perangkat lunak pengembangan karena menyediakan fasilitas yang memadai untuk membuat perangkat lunak yang berbasis *web* dan juga menyediakan *web server* yaitu Apache *web server*, database server yaitu Mysql *server*. PHP

dan *dreamweaver* 8 dipilih sebagai perangkat lunak pengembang karena menyediakan fasilitas yang memadai dan membuat perangkat lunak yang berbasis *web*. Sementara itu MySQL digunakan sebagai pengembang dalam pembuatan basis data.

### 4.4.3 Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain :

1. Pada sisi *server* perangkat keras yang dibutuhkan antara lain :
    a. Menggunakan prosessor Minimal Intel dual core atau yang sekelasnya.
    b. Menggunakan minimal RAM 1 GB.
    c. *Harddisk* untuk penampungan data minimal 40 Ghz.
    d. *Mouse, Keyboard* dan *monitor* sebagai peralatan antar muka.
    e. Kartu Jaringan atau modem

2. Pada sisi *client* perangkat keras yang dibutuhkan antara lain :
    a. Processor Minimal Intel Pentium 4 atau yang sekelasnya.
    b. Menggunakan minimal RAM 512 MB.
    c. Hardisk 40 Gb atau lebih.
    d. *Input Device* ( *Keyboard, Mouse* )
    e. *Output Device* ( Monitor, Printer )
    f. Kartu Jaringan atau modem

g. Sistem operasi ( Windows atau Linux ) yang mendukung aplikasi *browser*

### 4.4.4 Implementasi Basis Data (Sintaks SQL)

Implementasi basis data dilakukan dengan menggunakan dahasa SQL, dimana aplikasi pemrograman yang digunakan adalah MySQL, implementasi basis datanya dalam bahas SQL adalah sebagai berikut :

```
--
-- Database: `db_antz`
--
-- --------------------------------------------------------
--
-- Table structure for table `temp`
--
CREATE TABLE IF NOT EXISTS `temp` (
  `id_temp` int(11) NOT NULL AUTO_INCREMENT,
  `id_session` char(32) DEFAULT NULL,
  `id_subproduk` int(11) DEFAULT NULL,
  `qty` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_temp`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8
;
--
```

```
-- Dumping data for table `temp`

--

-- --------------------------------------------------------

--

-- Table structure for table `temp_layanan`

--

CREATE TABLE IF NOT EXISTS `temp_layanan` (

  `id_temp_layanan` int(11) NOT NULL AUTO_INCREMENT,

  `id_kurir` int(11) DEFAULT NULL,

  `id_layanan` int(11) DEFAULT NULL,

  `id_detail_kurir` int(11) DEFAULT NULL,

  `ongkos_kirim` int(11) DEFAULT NULL,

  `session_id` char(32) DEFAULT NULL,

  PRIMARY KEY (`id_temp_layanan`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=6

;


--

-- Dumping data for table `temp_layanan`

--

-- --------------------------------------------------------

--
```

```
-- Table structure for table `t_admin`
--
CREATE TABLE IF NOT EXISTS `t_admin` (
  `id_admin` int(11) NOT NULL AUTO_INCREMENT,
  `nama_admin` varchar(50) NOT NULL,
  `level_admin` int(11) NOT NULL,
  `email_admin` varchar(50) NOT NULL,
  `password_admin` char(41) NOT NULL,
  `phone` varchar(12) NOT NULL,
  `status_admin` int(11) NOT NULL,
  `insert_date` datetime NOT NULL,
  PRIMARY KEY (`id_admin`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4
;
--
-- Dumping data for table `t_admin`
--

INSERT INTO `t_admin` (`id_admin`, `nama_admin`, `level_admin`,
`email_admin`, `password_admin`, `phone`, `status_admin`, `insert_date`)
VALUES
```

```
(3, 'admin', 1, 'admin@antz.com',
'*4ACFE3202A5FF5CF467898FC58AAB1D615029441', '081322546378', 1,
'2013-06-13 05:33:57');
-- --------------------------------------------------------
--
-- Table structure for table `t_detail_kurir`
--
CREATE TABLE IF NOT EXISTS `t_detail_kurir` (
  `id_detail_kurir` int(11) NOT NULL AUTO_INCREMENT,
  `id_layanan` int(11) DEFAULT NULL,
  `id_kota` int(11) DEFAULT NULL,
  `harga_kirim` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_detail_kurir`),
  KEY `FK_layanan` (`id_layanan`),
  KEY `FK_detailkota` (`id_kota`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=FIXED AUTO_INCREMENT=10 ;
--
-- Dumping data for table `t_detail_kurir`
--
INSERT INTO `t_detail_kurir` (`id_detail_kurir`, `id_layanan`, `id_kota`,
`harga_kirim`) VALUES
```

```
(1, 1, 4, 10000),

(2, 2, 4, 9000),

(3, 2, 1, 3000),

(4, 1, 1, 3000),

(5, 1, 2, 8000),

(6, 3, 2, 5000),

(7, 4, 2, 6500),

(8, 4, 5, 11500),

(9, 1, 5, 20500);

-- --------------------------------------------------

--

-- Table structure for table `t_gambar`

-

CREATE TABLE IF NOT EXISTS `t_gambar` (

  `id_gambar` int(11) NOT NULL AUTO_INCREMENT,

  `kode_produk` varchar(10) DEFAULT NULL,

  `gambar` varchar(100) DEFAULT NULL,

  `direktori` varchar(100) DEFAULT NULL,

  PRIMARY KEY (`id_gambar`),

  KEY `FK_t_gambar` (`kode_produk`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9

;
```

```
--

-- Dumping data for table `t_gambar`

--

INSERT INTO `t_gambar` (`id_gambar`, `kode_produk`, `gambar`,
`direktori`) VALUES

(1, 'SWAFG011W', '8750jaket 1.jpg', 'produk/8750jaket 1.jpg'),

(2, 'SWLKS012P', '4272sweater-1.jpg', 'produk/4272sweater-1.jpg'),

(3, 'SPADO014P', '624.JPG', 'produk/624.JPG'),

(4, 'SPVNS016P', '9517.jpg', 'produk/9517.jpg'),

(5, 'BKAFG017P', '431.jpg', 'produk/431.jpg'),

(6, 'BKSK019P', '112.jpg', 'produk/112.jpg'),

(7, 'BKSK020P', '603.jpg', 'produk/603.jpg'),

(8, 'BKSK021P', '271baj 6.jpg', 'produk/271baj 6.jpg');

-- --------------------------------------------------------

--

-- Table structure for table `t_kategori`

--

CREATE TABLE IF NOT EXISTS `t_kategori` (

  `kode_kategori` char(3) NOT NULL,

  `kategori` varchar(20) DEFAULT NULL,

  PRIMARY KEY (`kode_kategori`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC;
--
-- Dumping data for table `t_kategori`
--
INSERT INTO `t_kategori` (`kode_kategori`, `kategori`) VALUES
('BK', 'Baju'),
('JC', 'Jeans'),
('SP', 'Sepatu'),
('SW', 'Sweater');
-- --------------------------------------------------------
--
-- Table structure for table `t_kota`
--
CREATE TABLE IF NOT EXISTS `t_kota` (
  `id_kota` int(11) NOT NULL AUTO_INCREMENT,
  `id_provinsi` int(11) DEFAULT NULL,
  `kota` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_kota`),
  KEY `FK_provinsi` (`id_provinsi`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC
AUTO_INCREMENT=7 ;
--
-- Dumping data for table `t_kota`
--
INSERT INTO `t_kota` (`id_kota`, `id_provinsi`, `kota`) VALUES
(1, 1, 'Bandung'),
(2, 1, 'Majalengka'),
(3, 1, 'Karawang'),
(4, 2, 'Jakarta Pusat'),
(5, 2, 'Jakarta Selatan'),
(6, 2, 'Jakarta Utara');
-- --------------------------------------------------------
--
-- Table structure for table `t_kurir`
--
CREATE TABLE IF NOT EXISTS `t_kurir` (
  `id_kurir` int(11) NOT NULL AUTO_INCREMENT,
  `nama_kurir` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id_kurir`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=DYNAMIC AUTO_INCREMENT=3 ;
--
-- Dumping data for table `t_kurir`
--
INSERT INTO `t_kurir` (`id_kurir`, `nama_kurir`) VALUES
(1, 'TIKI JNE'),
(2, 'POS');
-- --------------------------------------------------------
--
-- Table structure for table `t_layanan`
--
CREATE TABLE IF NOT EXISTS `t_layanan` (
  `id_layanan` int(11) NOT NULL AUTO_INCREMENT,
  `nama_layanan` varchar(20) DEFAULT NULL,
  `id_kurir` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_layanan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=DYNAMIC AUTO_INCREMENT=5 ;
--
-- Dumping data for table `t_layanan`
--
```

```
INSERT INTO `t_layanan` (`id_layanan`, `nama_layanan`, `id_kurir`) VALUES
(1, 'Regullar', 1),
(2, 'Ons', 1),
(3, 'Khusus', 2),
(4, 'Kilat Khusus', 2);
-- --------------------------------------------------------
--
-- Table structure for table `t_layanan_pembayaran`
--
CREATE TABLE IF NOT EXISTS `t_layanan_pembayaran` (
  `id_layananpembayaran` int(11) NOT NULL AUTO_INCREMENT,
  `nama_bank` varchar(20) NOT NULL,
  `data_bank` text NOT NULL,
  `image` varchar(100) NOT NULL,
  PRIMARY KEY (`id_layananpembayaran`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;
--
-- Dumping data for table `t_layanan_pembayaran`
--
```

```
INSERT INTO `t_layanan_pembayaran` (`id_layananpembayaran`, `nama_bank`, `data_bank`, `image`) VALUES
(10, 'BNI', 'a/n. Lucky Januar, \r\nNomor Rekening : 0999909', '85bni.jpeg'),
(11, 'MANDIRI', 'a/s. Lucky Januar, \r\nNomor Rekening : 09989088', '99mandiri.jpg'),
(12, 'PAYPAL', 'Lucky12@gmail.com', '29paypal.jpeg');
-- --------------------------------------------------------
--
-- Table structure for table `t_member`
--
CREATE TABLE IF NOT EXISTS `t_member` (
  `id_member` int(11) NOT NULL AUTO_INCREMENT,
  `nama_depan` varchar(30) DEFAULT NULL,
  `nama_belakang` varchar(30) DEFAULT NULL,
  `alamat_lengkap` text,
  `email_member` varchar(30) DEFAULT NULL,
  `password_member` varchar(41) DEFAULT NULL,
  `id_kota` int(11) DEFAULT NULL,
  `phone` varchar(12) DEFAULT NULL,
  `insert_date` datetime DEFAULT NULL,
  `id_pertanyaan` int(11) DEFAULT NULL,
  `jawaban` varchar(20) DEFAULT NULL,
```

```
  `register_session_id` varchar(32) DEFAULT NULL,
  `last_login` datetime DEFAULT NULL,
  `status_member` int(11) NOT NULL,
  `is_active` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_member`),
  UNIQUE KEY `email_member` (`email_member`),
  KEY `FK_kota` (`id_kota`),
  KEY `FK_pertanyaan` (`id_pertanyaan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC
AUTO_INCREMENT=4 ;


--
-- Dumping data for table `t_member`
--
INSERT INTO `t_member` (`id_member`, `nama_depan`, `nama_belakang`,
`alamat_lengkap`, `email_member`, `password_member`, `id_kota`, `phone`,
`insert_date`, `id_pertanyaan`, `jawaban`, `register_session_id`, `last_login`,
`status_member`, `is_active`) VALUES
(1, 'xxx', 'xxx', 'xxx', 'xxx@gmail.com',
'*E6CC90B878B948C35E92B003C792C46C58C4AF40', 5, '123', '2013-06-
```

```
18 01:31:48', 1, 'y', '97641b7d4541c9995d4edacca14d977e', '2013-06-18 01:43:56', 1, 1),
(2, 'singgih', 'wahyu', 'sukarajin 2', 'singgihw12@gmail.com', '*E6CC90B878B948C35E92B003C792C46C58C4AF40', 4, '0223', '2013-06-19 15:15:13', 1, 'dudung', 'd40948c94e8e8b900be6b793da121c2b', '2013-06-24 04:51:32', 1, 1),
(3, 'singgih', 'wahyu', 'sukarajin', 'singgih@gmail.com', '*E6CC90B878B948C35E92B003C792C46C58C4AF40', 2, '09876', '2013-06-19 17:05:42', 1, 'dudung', '53d152a1c0ba6fcdb94ef1515701d198', '2013-06-19 17:07:12', 1, 1);
-- --------------------------------------------------------
--
-- Table structure for table `t_merk`
--
CREATE TABLE IF NOT EXISTS `t_merk` (
  `kode_merk` varchar(5) NOT NULL,
  `merk` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`kode_merk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=DYNAMIC;

--
```

```
-- Dumping data for table `t_merk`
--
INSERT INTO `t_merk` (`kode_merk`, `merk`) VALUES
('AD', 'Adidas'),
('ADO', 'Adio'),
('AFG', 'After Glow'),
('LKS', 'Lakers'),
('NK', 'Nike'),
('SK', 'Sketer'),
('VNS', 'Vans');
-- --------------------------------------------------------
--
-- Table structure for table `t_model`
--

CREATE TABLE IF NOT EXISTS `t_model` (
  `id_model` int(11) NOT NULL AUTO_INCREMENT,
  `kode_kategori` char(3) DEFAULT NULL,
  `model` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id_model`),
  KEY `FK_kategori` (`kode_kategori`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC
AUTO_INCREMENT=12 ;

--
-- Dumping data for table `t_model`
--

INSERT INTO `t_model` (`id_model`, `kode_kategori`, `model`) VALUES
(6, 'BK', 'Lengan Pendek'),
(7, 'BK', 'Lengan Panjang'),
(8, 'JC', 'Jeans Panjang'),
(9, 'JC', 'Jeans Pendek'),
(10, 'SW', 'Sweater Panjang'),
(11, 'SP', 'Sport');

-- --------------------------------------------------------

--
-- Table structure for table `t_order`
--
```

```
CREATE TABLE IF NOT EXISTS `t_order` (
  `id_order` int(11) NOT NULL AUTO_INCREMENT,
  `id_member` int(11) DEFAULT NULL,
  `total_qty` int(11) DEFAULT NULL,
  `total_berat` float DEFAULT NULL,
  `total_ongkos` int(11) DEFAULT NULL,
  `total_bayar` int(11) DEFAULT NULL,
  `tanggal_order` date DEFAULT NULL,
  `tanggal_expired` date NOT NULL,
  `status_order` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_order`),
  KEY `FK_member` (`id_member`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=FIXED AUTO_INCREMENT=12 ;
--
-- Dumping data for table `t_order`
--
INSERT INTO `t_order` (`id_order`, `id_member`, `total_qty`, `total_berat`,
`total_ongkos`, `total_bayar`, `tanggal_order`, `tanggal_expired`,
`status_order`) VALUES
(1, 0, 1, 1, 9000, 79000, '2013-06-18', '2013-06-28', 1),
(2, 1, 1, 1, 5000, 77000, '2013-06-18', '2013-06-28', 1),
```

```
(3, 0, 2, 1, 3000, 143000, '2013-06-19', '2013-06-29', 1),
(4, 2, 1, 1, 3000, 73000, '2013-06-19', '2013-06-29', 1),
(5, 0, 1, 1, 5000, 75000, '2013-06-19', '2013-06-29', 1),
(6, 3, 2, 2, 6000, 266000, '2013-06-19', '2013-06-29', 1),
(7, 2, 2, 2, 5000, 275000, '2013-06-24', '2013-07-04', 1),
(8, 2, 1, 1, 3000, 73000, '2013-06-24', '2013-07-04', 0),
(9, 2, 3, 1.5, 3000, 213000, '2013-06-24', '2013-07-04', 1),
(10, 2, 3, 2.5, 22500, 352500, '2013-06-24', '2013-07-04', 1),
(11, 2, 1, 1, 6500, 76500, '2013-06-24', '2013-07-04', 1);
-- --------------------------------------------------------
--
-- Table structure for table `t_order_detail`
--
CREATE TABLE IF NOT EXISTS `t_order_detail` (
  `id_order_detail` int(11) NOT NULL AUTO_INCREMENT,
  `id_order` int(11) DEFAULT NULL,
  `id_subproduk` int(11) DEFAULT NULL,
  `qty_order` int(11) DEFAULT NULL,
  `berat_order` float DEFAULT NULL,
  `diskon_order` int(11) DEFAULT NULL,
  `harga_order` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_order_detail`),
```

```
  KEY `FK_order` (`id_order`),
  KEY `FK_t_order_detail` (`id_subproduk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=FIXED AUTO_INCREMENT=15 ;

--
-- Dumping data for table `t_order_detail`
--
INSERT INTO `t_order_detail` (`id_order_detail`, `id_order`, `id_subproduk`, `qty_order`, `berat_order`, `diskon_order`, `harga_order`) VALUES
(1, 1, 3, 1, 1, 0, 70000),
(2, 2, 1, 1, 0.5, 10, 80000),
(3, 3, 6, 2, 0.5, 0, 70000),
(4, 4, 8, 1, 0.5, 0, 70000),
(5, 5, 7, 1, 0.5, 0, 70000),
(6, 6, 4, 1, 1.5, 5, 200000),
(7, 6, 8, 1, 0.5, 0, 70000),
(8, 7, 8, 1, 0.5, 0, 70000),
(9, 7, 5, 1, 1.5, 0, 200000),
(10, 8, 3, 1, 1, 0, 70000),
(11, 9, 8, 3, 0.5, 0, 70000),
(12, 10, 8, 2, 0.5, 0, 70000),
```

```
(13, 10, 4, 1, 1.5, 5, 200000),

(14, 11, 8, 1, 0.5, 0, 70000);


-- --------------------------------------------------------


--

-- Table structure for table `t_pembayaran`

--


CREATE TABLE IF NOT EXISTS `t_pembayaran` (

  `id_pembayaran` int(11) NOT NULL AUTO_INCREMENT,

  `id_order` int(11) DEFAULT NULL,

  `jenis_pembayaran` varchar(20) DEFAULT NULL,

  `bank_tujuan` varchar(15) NOT NULL,

  `nama_layanan_pembayaran` varchar(20) DEFAULT NULL,

  `atas_nama` varchar(50) DEFAULT NULL,

  `no_rekening` varchar(20) DEFAULT NULL,

  `jumlah_pembayaran` float DEFAULT NULL,

  `tanggal_bayar` date DEFAULT NULL,

  `status_pembayaran` int(11) DEFAULT NULL,

  PRIMARY KEY (`id_pembayaran`),

  KEY `FK_pembayaranorder` (`id_order`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=DYNAMIC AUTO_INCREMENT=12 ;

--
-- Dumping data for table `t_pembayaran`
--

INSERT INTO `t_pembayaran` (`id_pembayaran`, `id_order`,
`jenis_pembayaran`, `bank_tujuan`, `nama_layanan_pembayaran`,
`atas_nama`, `no_rekening`, `jumlah_pembayaran`, `tanggal_bayar`,
`status_pembayaran`) VALUES
(1, 1, 'Transfer', 'BCA', 'BCA', 'XXX', '12345678', 79000, '2013-06-18', 1),
(2, 2, 'Transfer', 'BRI', 'BRI', 'XXX', '12345', 77000, '2013-06-18', 1),
-- --------------------------------------------------------

--
-- Table structure for table `t_pengiriman`
--

CREATE TABLE IF NOT EXISTS `t_pengiriman` (
  `id_pengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_order` int(11) DEFAULT NULL,
```

```
  `id_kota` int(11) DEFAULT NULL,
  `id_detail_kurir` int(11) DEFAULT NULL,
  `tanggal_kirim` datetime DEFAULT NULL,
  `nama_penerima` varchar(50) DEFAULT NULL,
  `alamat_penerima` text,
  `email` varchar(50) DEFAULT NULL,
  `telp` varchar(12) DEFAULT NULL,
  `status_kirim` int(11) DEFAULT '0',
  `status_terima` int(11) NOT NULL,
  `ongkos_kirim` int(11) DEFAULT NULL,
  `no_resi` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id_pengiriman`),
  KEY `FK_t_pengiriman` (`id_detail_kurir`),
  KEY `FK_orderpengiriman` (`id_order`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=DYNAMIC AUTO_INCREMENT=12 ;

--
-- Dumping data for table `t_pengiriman`
--
INSERT INTO `t_pengiriman` (`id_pengiriman`, `id_order`, `id_kota`,
`id_detail_kurir`, `tanggal_kirim`, `nama_penerima`, `alamat_penerima`,
```

```
`email`, `telp`, `status_kirim`, `status_terima`, `ongkos_kirim`, `no_resi`) VALUES
(1, 1, 4, 2, '2013-06-18 01:37:44', 'xxx xxx', 'xxx', 'xxx@gmail.com', '123', 1, 1, 9000, ''),
(2, 2, 2, 6, '2013-06-18 01:47:02', 'xxx xxx', 'xxx', 'xxx@gmail.com', '123', 1, 1, 5000, '2'),
(3, 3, 1, 3, '2013-06-19 15:17:41', 'singgih wahyu', 'sukarajin 2', 'singgihw12@gmail.com', '0223', 1, 1, 3000, '789'),
-- --------------------------------------------------------

--
-- Table structure for table `t_pertanyaan`
--

CREATE TABLE IF NOT EXISTS `t_pertanyaan` (
  `id_pertanyaan` int(11) NOT NULL AUTO_INCREMENT,
  `pertanyaan` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_pertanyaan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

--
```

```
-- Dumping data for table `t_pertanyaan`
--

INSERT INTO `t_pertanyaan` (`id_pertanyaan`, `pertanyaan`) VALUES
(1, 'Siapa guru pertama Anda?'),
(2, 'Hewan peliharaan kesayangan Anda?');

-- --------------------------------------------------------

--
-- Table structure for table `t_produk`
--

CREATE TABLE IF NOT EXISTS `t_produk` (
  `kode_produk` varchar(10) NOT NULL,
  `no_urut` int(3) unsigned zerofill NOT NULL AUTO_INCREMENT,
  `kode_merk` varchar(5) DEFAULT NULL,
  `id_model` int(11) DEFAULT NULL,
  `jenis_produk` char(1) DEFAULT NULL,
  `nama_produk` varchar(30) DEFAULT NULL,
  `tanggal_masuk` datetime DEFAULT NULL,
  `deskripsi` tinytext,
```

```
  `diskon` int(11) DEFAULT NULL,
  PRIMARY KEY (`kode_produk`),
  UNIQUE KEY `no_urut` (`no_urut`),
  KEY `FK_merk` (`kode_merk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=DYNAMIC AUTO_INCREMENT=23 ;

--
-- Dumping data for table `t_produk`
--

INSERT INTO `t_produk` (`kode_produk`, `no_urut`, `kode_merk`, `id_model`, `jenis_produk`, `nama_produk`, `tanggal_masuk`, `deskripsi`, `diskon`) VALUES
('BKAFG017P', 018, 'AFG', 7, 'P', 'Baju001', '2011-03-23 19:05:40', 'Keren abizzzzzzzzzzzzzzzz..', 0),
('BKSK019P', 020, 'SK', 7, 'P', 'Baju 20', '2011-03-30 13:33:13', '', 0),
('BKSK020P', 021, 'SK', 7, 'P', 'Baju 21', '2011-03-30 13:33:20', '', 0),
('BKSK021P', 022, 'SK', 7, 'P', 'Baju 22', '2011-03-30 13:33:24', '', 0),
('SPADO014P', 016, 'SK', 11, 'P', 'Sepatu 001', '2011-03-11 19:30:00', 'Kuat, nyaman, style anak muda', 5),
```

('SPVNS016P', 017, 'VNS', 11, 'P', 'Spantu vans002', '2011-03-11 19:36:50', '', 0),

('SWAFG011W', 012, 'SK', 10, 'W', 'Sweater001CW', '2011-03-10 13:49:18', 'Keran dan cocok untuk anak muda, menggunakan bahan yang halus', 10),

('SWLKS012P', 014, 'SK', 10, 'P', 'Lakers 001', '2011-03-11 19:18:35', 'Pokoke matab lah', 0),

('SWLKS018P', 019, 'LKS', 10, 'P', 'test000001', '2011-03-30 13:32:20', 'keran dan nyaman', 0);

-- --------------------------------------------------------

--

-- Table structure for table `t_provinsi`

--

```
CREATE TABLE IF NOT EXISTS `t_provinsi` (
  `id_provinsi` int(11) NOT NULL AUTO_INCREMENT,
  `provinsi` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_provinsi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 CHECKSUM=1 DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC AUTO_INCREMENT=3 ;
```

```
--

-- Dumping data for table `t_provinsi`

--


INSERT INTO `t_provinsi` (`id_provinsi`, `provinsi`) VALUES

(1, 'Jawa Barat'),

(2, 'DKI Jakarta');


-- --------------------------------------------------------


--

-- Table structure for table `t_retur`

--


CREATE TABLE IF NOT EXISTS `t_retur` (

  `id_retur` int(11) NOT NULL AUTO_INCREMENT,

  `id_order` int(11) NOT NULL,

  `id_subproduk` int(11) NOT NULL,

  `jumlah_retur` int(11) NOT NULL,

  `status_retur` int(11) NOT NULL,

  `tanggal_retur` date NOT NULL,
```

```
  `no_resi` varchar(20) NOT NULL,
  PRIMARY KEY (`id_retur`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4
;

--
-- Dumping data for table `t_retur`
--

INSERT INTO `t_retur` (`id_retur`, `id_order`, `id_subproduk`,
`jumlah_retur`, `status_retur`, `tanggal_retur`, `no_resi`) VALUES
(1, 2, 1, 1, 2, '2013-06-18', '5'),
(2, 4, 8, 1, 2, '2013-06-19', '123'),
(3, 6, 8, 1, 2, '2013-06-19', '567');

-- --------------------------------------------------------

--
-- Table structure for table `t_subproduk`
--

CREATE TABLE IF NOT EXISTS `t_subproduk` (
```

```
  `id_subproduk` int(11) NOT NULL AUTO_INCREMENT,
  `kode_produk` varchar(10) DEFAULT NULL,
  `id_ukuran` int(11) DEFAULT NULL,
  `id_warna` int(11) DEFAULT NULL,
  `berat` float DEFAULT NULL,
  `harga_dasar` int(11) DEFAULT NULL,
  `harga_jual` int(11) DEFAULT NULL,
  `laba` float DEFAULT NULL,
  `stok` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_subproduk`),
  KEY `FK_ukuran` (`id_ukuran`),
  KEY `FK_warna` (`id_warna`),
  KEY `FK_t_subproduk` (`kode_produk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
ROW_FORMAT=FIXED AUTO_INCREMENT=10 ;

--
-- Dumping data for table `t_subproduk`
--

INSERT INTO `t_subproduk` (`id_subproduk`, `kode_produk`, `id_ukuran`,
`id_warna`, `berat`, `harga_dasar`, `harga_jual`, `laba`, `stok`) VALUES
```

```
(1, 'SWAFG011W', 1, 6, 0.5, 50000, 80000, 30000, 5),
(2, 'SWAFG011W', 3, 6, 0.5, 50000, 80000, 30000, 5),
(3, 'SWLKS012P', 3, 2, 1, 50000, 70000, 20000, 8),
(4, 'SPADO014P', 4, 4, 1.5, 100000, 200000, 100000, 4),
(5, 'SPVNS016P', 4, 6, 1.5, 100000, 200000, 100000, 3),
(6, 'BKAFG017P', 3, 6, 0.5, 50000, 70000, 20000, 0),
(7, 'BKSK019P', 3, 6, 0.5, 50000, 70000, 20000, 2),
(8, 'BKSK020P', 3, 6, 0.5, 50000, 70000, 20000, 2),
(9, 'BKSK021P', 3, 3, 0.5, 50000, 70000, 20000, 5);

-- --------------------------------------------------------

--
-- Table structure for table `t_ukuran`
--

CREATE TABLE IF NOT EXISTS `t_ukuran` (
  `id_ukuran` int(11) NOT NULL AUTO_INCREMENT,
  `ukuran` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`id_ukuran`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 CHECKSUM=1
DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC
AUTO_INCREMENT=5 ;

--
-- Dumping data for table `t_ukuran`
--

INSERT INTO `t_ukuran` (`id_ukuran`, `ukuran`) VALUES
(1, 'S'),
(2, 'XL'),
(3, 'L'),
(4, '40');

-- --------------------------------------------------------

--
-- Table structure for table `t_warna`
--

CREATE TABLE IF NOT EXISTS `t_warna` (
  `id_warna` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `warna` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id_warna`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=DYNAMIC AUTO_INCREMENT=7 ;

--
-- Dumping data for table `t_warna`
--

INSERT INTO `t_warna` (`id_warna`, `warna`) VALUES
(2, 'Kuning'),
(3, 'Hijau'),
(4, 'Hitam'),
(5, 'Putih'),
(6, 'Merah');

--
-- Constraints for dumped tables
--

--
-- Constraints for table `t_detail_kurir`
```

```
--
ALTER TABLE `t_detail_kurir`
 ADD CONSTRAINT `FK_detailkota` FOREIGN KEY (`id_kota`) REFERENCES `t_kota` (`id_kota`),
 ADD CONSTRAINT `FK_layanan` FOREIGN KEY (`id_layanan`) REFERENCES `t_layanan` (`id_layanan`);

--
-- Constraints for table `t_gambar`
--
ALTER TABLE `t_gambar`
 ADD CONSTRAINT `FK_t_gambar` FOREIGN KEY (`kode_produk`) REFERENCES `t_produk` (`kode_produk`) ON DELETE CASCADE;

--
-- Constraints for table `t_kota`
--
ALTER TABLE `t_kota`
 ADD CONSTRAINT `FK_provinsi` FOREIGN KEY (`id_provinsi`) REFERENCES `t_provinsi` (`id_provinsi`);

--
```

```
-- Constraints for table `t_member`
--
ALTER TABLE `t_member`
 ADD CONSTRAINT `FK_kota` FOREIGN KEY (`id_kota`) REFERENCES `t_kota` (`id_kota`),
 ADD CONSTRAINT `FK_pertanyaan` FOREIGN KEY (`id_pertanyaan`) REFERENCES `t_pertanyaan` (`id_pertanyaan`);

--
-- Constraints for table `t_model`
--
ALTER TABLE `t_model`
 ADD CONSTRAINT `FK_kategori` FOREIGN KEY (`kode_kategori`) REFERENCES `t_kategori` (`kode_kategori`);

--
-- Constraints for table `t_order_detail`
--
ALTER TABLE `t_order_detail`
 ADD CONSTRAINT `FK_order` FOREIGN KEY (`id_order`) REFERENCES `t_order` (`id_order`) ON DELETE CASCADE,
```

```
 ADD CONSTRAINT `FK_t_order_detail` FOREIGN KEY (`id_subproduk`) REFERENCES `t_subproduk` (`id_subproduk`);

--
-- Constraints for table `t_pembayaran`
--
ALTER TABLE `t_pembayaran`
 ADD CONSTRAINT `FK_pembayaranorder` FOREIGN KEY (`id_order`) REFERENCES `t_order` (`id_order`);

--
-- Constraints for table `t_pengiriman`
--
ALTER TABLE `t_pengiriman`
 ADD CONSTRAINT `FK_orderpengiriman` FOREIGN KEY (`id_order`) REFERENCES `t_order` (`id_order`) ON DELETE CASCADE,
 ADD CONSTRAINT `FK_t_pengiriman` FOREIGN KEY (`id_detail_kurir`) REFERENCES `t_detail_kurir` (`id_detail_kurir`);

--
-- Constraints for table `t_produk`
--
```

```
ALTER TABLE `t_produk`
 ADD CONSTRAINT `FK_merk` FOREIGN KEY (`kode_merk`) REFERENCES `t_merk` (`kode_merk`);


--
-- Constraints for table `t_subproduk`
--
ALTER TABLE `t_subproduk`
 ADD CONSTRAINT `FK_t_subproduk` FOREIGN KEY (`kode_produk`) REFERENCES `t_produk` (`kode_produk`),
 ADD CONSTRAINT `FK_ukuran` FOREIGN KEY (`id_ukuran`) REFERENCES `t_ukuran` (`id_ukuran`),
 ADD CONSTRAINT `FK_warna` FOREIGN KEY (`id_warna`) REFERENCES `t_warna` (`id_warna`);
```

### 4.4.5 Implementasi Antar Muka

Dalam PHP (4.1.1), implementasi antarmuka dilakukan dengan sebuah modul *web page* yang berekstensi PHP. Setiap halaman dalam perangkat lunak dibuat *File* program yang ditulis dengan ekstensi php. Berikut ini adalah implementasi dari setiap halaman yang dibuat dan dibedakan berdasarkan Kebutuhan User dan Administrator (Petugas) AntzShop.

#### 4.4.5.1 Implementasi Halaman Utama *User*

Implementasi halaman yang terdapat pada *user* yang dibuat bentuk file program dengan berekstensi PHP dapat dilihat pada tabel 5.1 dan tabel 5.2.

**Tabel 5.1** Implementasi Halaman Utama *User Non Member*.

| Menu | Deskripsi | Nama File |
|---|---|---|
| Home | File program untuk menangani halaman utama. | Index.php |
| Cara pemesanan | File program yang menangani untuk mengetahui cara pemesanan produk. | Carapesan.php |
| Tentang Kami | File program untuk menangani informasi tentang profil perusahaan. | Aboutus.php |
| Cara pembayaran | File program yang menangani informasi tentang cara melakukan pembayaran | Carabayar.php |
| Sign in | File program yang menangani login untuk member. | Login.php |
| Register | File program yang menangani untuk registrasi member. | Register.php |

**Tabel 5.2** Implementasi Halaman Utama User *Member*.

| Menu | Deskripsi | Nama File |
|---|---|---|
| Home | Menampilkan untuk menangani halaman utama | Indek.php |
| Cara pemesanan | File program yang menangani untuk mengetahui cara pemesanan produk. | carapesan.php |
| Cara pembayaran | File program yang | carabayar.php |

| | | |
|---|---|---|
| | menangani informasi tentang cara melakukan pembayaran. | |
| Tentang kami | File program untuk menangani informasi tentang profil perusahaan. | Aboutus.php |
| History | File program untuk menangani informasi tentang proses transaksi pemesanan. | History.php |
| Profile | File program yang menangani profil member | Profile.php |
| Ganti password | File program yang menangani untuk ganti password member | Gpaa.php |
| Retur | File program yang menangani retur barang | Retur.php |

#### 4.4.5.2 Implementasi Halaman Petugas (Admin)

Implementasi halaman yang terdapat *petugas* yang dibuat bentuk *file* program dengan berekstensi PHP dapat dilihat pada tabel 5.3.

**Tabel 5.3** Implementasi Halaman Utama Petugas

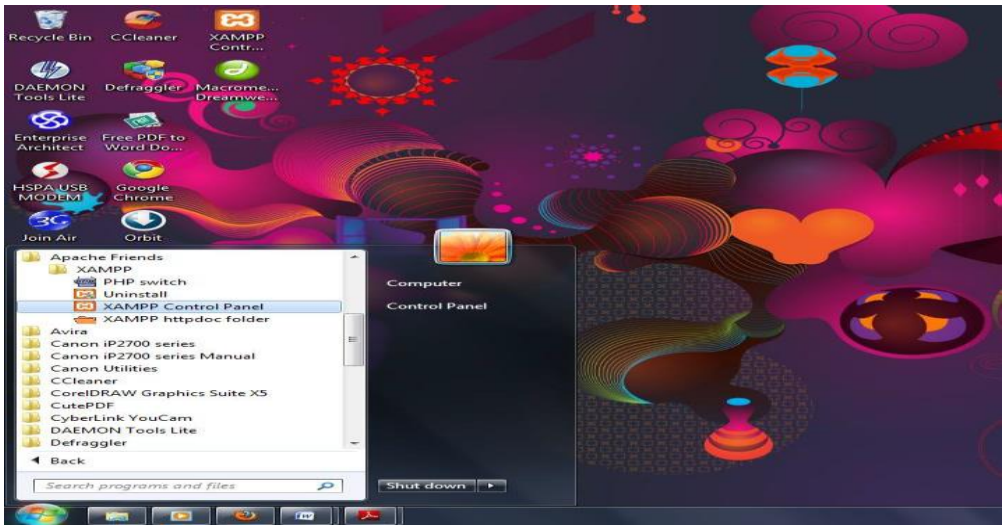
| Menu | Deskripsi | Nama file |
|---|---|---|
| Home | File program yang menangani halaman utama | Jendelaadmin.php |
| Admin | File program yang menangani akses Admin petugas | Admin.php |
| Member | File program yang menangani informasi data member | Member.php |
| Kategori | File program yang menangani jenis produk | Kategori.php |
| Model | File program yang menangani jenis | Model.php |

| | | |
|---|---|---|
| | produk berdasarkan model | |
| Merk | File program yang menangani jenis merk produk | Merk.php |
| Ukuran | File program yang menangani jenis produk berdasarkan ukuran | Ukuran.php |
| Warna | File program yang menangani jenis produk berdasarkan warna | Warna.php |
| Produk | File program yang menangani tambah produk | Produk.php |
| Provinsi | File program yang menangani pilihan provinsi antar tujuan | Provinsi.php |
| Kota | File program yang menangani pilihan kota antar tujuan | Kota.php |
| Kurir | File program yang menangani jasa pengiriman | Kurir.php |
| Layanan pengiriman | File program yang menangani jasa pengiriman berdasarkan kelas | Layanan.php |
| Ongkos kirim | File program yang menangani ongkos pengiriman | Ongkos.php |
| Transaksi | File program yang menangani pemesanan | Transaksi.php |
| Pesanan konfirmasi pembayaran | File program yang menangani transaksi yang sudah di bayar | Pesanan.php |
| Barang terkirim | File program yang menangani barang yang sudah di kirim | Konfirmasi.php |
| Retur barang | File program yang menangani retur produk | Retur.php |
| Laporan produk | File program yang menangani produk yang terjual | Laporanproduk.php |

### 4.4.6 Implementasi Instalasi Program

Berikut ini adalah implementasi instalasi program untuk menggunakan program Sistem Informasi Penjualan Pada Toko AntzShop Berbasis Web. Langkah-langkahnya adalah sebagai berikut :

a. Klik menu start kemudian pilih apachefriends, xampp, XAMPP Control Panel Application.




**Gambar 5.1** Mengaktifkan XAMPP Control Panel Application.

b. Aktifkan XAMPP Control Panel Application seperti pada gambar di bawah.




**Gambar 5.2** XAMPP Control Panel Application.

c. Buka Mozilla firefox, lalu ketikan alamat http://localhost/antzshop/




**Gambar 5.3** Tampilan URL

### 4.4.7 Penggunaan Program

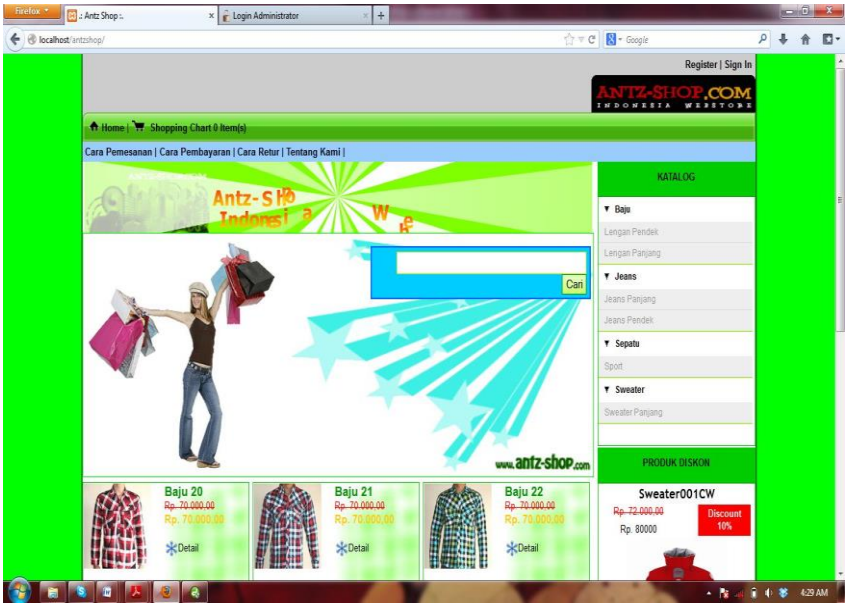
Penggunaan program merupakan kegiatan yang bertujuan untuk menerangkan secara singkat bagaimana penggunaan program Perancangan Sistem Informasi Penjualan pada Toko AntzShop. Adapun cara pengoperasiannya seperti yang akan dijelaskan berikut ini :

#### 4.4.7.1 Penggunan Program *user*

Berikut adalah petunjuk penggunaan program Sistem Informasi Penjualan Online AntzShop bagi *user*.

a. Home

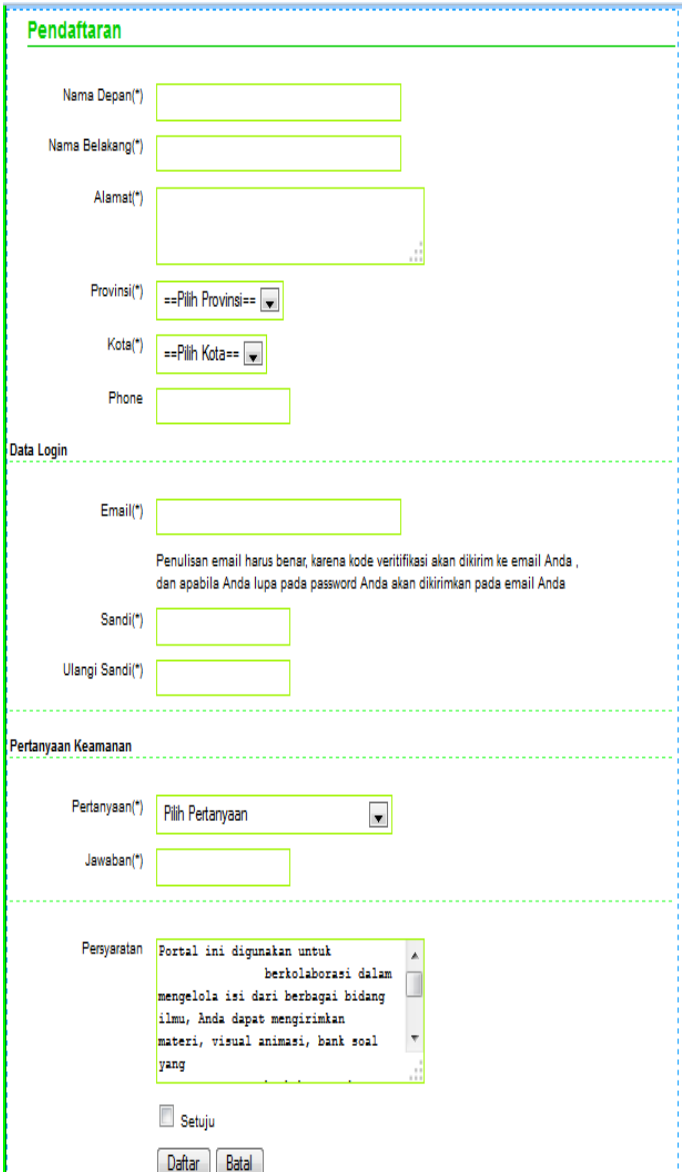
Tampilan Home adalah tampilan yang pertama kali muncul apabila *user* mengakses situs AntzShop. Adapun tampilannya adalah sebagai berikut:




**Gambar 5.4** Tampilan Halaman Utama AntzShop.

## 1. Tampilan Input Sebelum Regestrasi

a. Form Regestrasi

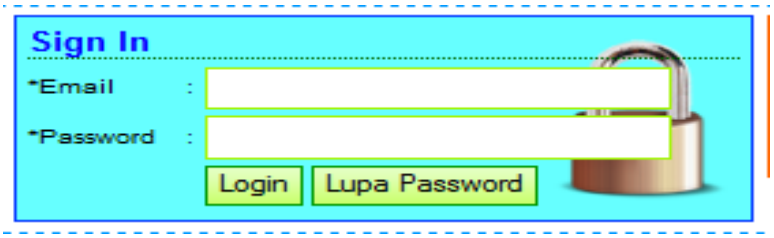



**Gambar 5.5** Tampilan Form Regestrasi Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

## 2. Tampilan Input Member

a. Form Login Member

Pada Tampilan login member, digunakan untuk member masuk kedalam halaman member untuk melakukan pemesanan produk.




**Gambar 5.6** Tampilan Login Member Perancangan sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web.*

b. Halaman Pemesanan Produk

Pada Tampilan Pemesanan Produk, digunakan member untuk melakukan pemesanan Produk.




**Gambar 5.7** Tampilan Form Pemesanan Produk Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web.*

b. Form Konfirmasi Pembayaran




**Gambar 5.8** Tampilan Form Konfirmasi Pembayaran Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web*.

c. Halaman Tampilan Produk

Dihalaman tampilan produk ini ditampilkan Produk yang tersedia di *website* AntzShop.




**Gambar 5.9** Tampilan Halaman Produk Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

d. Halaman Detail Produk

Dihalaman Produk ini ditampilkan Detail Produk di Website AntzShop

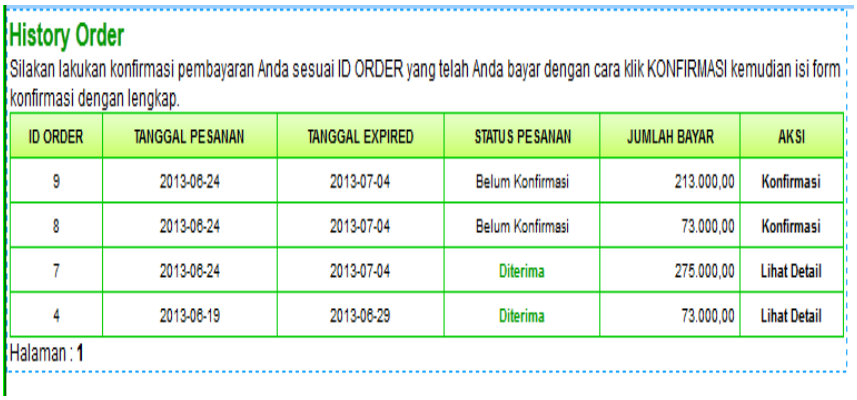



**Gambar 5.10** Tampilan Halaman Detail Produk Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web*.

**4. Tampilan Halaman Member**

a. Daftar Konfirmasi Pesanan

Pada halaman ini member dapat mengetahui data pemesanan produk yang diinginkan dan member dapat melakukan Konfirmasi Pembayaran.



History Order

Silakan lakukan konfirmasi pembayaran Anda sesuai ID ORDER yang telah Anda bayar dengan cara klik KONFIRMASI kemudian isi form konfirmasi dengan lengkap.

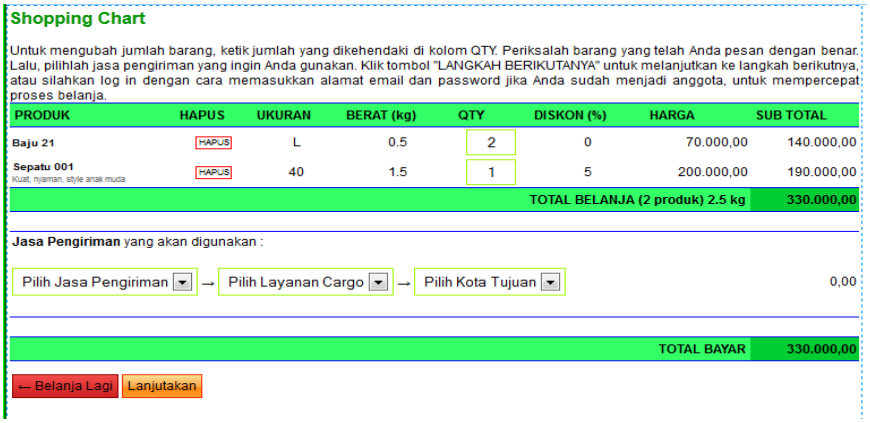
| ID ORDER | TANGGAL PESANAN | TANGGAL EXPIRED | STATUS PESANAN | JUMLAH BAYAR | AKSI |
|---|---|---|---|---|---|
| 9 | 2013-06-24 | 2013-07-04 | Belum Konfirmasi | 213.000,00 | Konfirmasi |
| 8 | 2013-06-24 | 2013-07-04 | Belum Konfirmasi | 73.000,00 | Konfirmasi |
| 7 | 2013-06-24 | 2013-07-04 | Diterima | 275.000,00 | Lihat Detail |
| 4 | 2013-06-19 | 2013-06-29 | Diterima | 73.000,00 | Lihat Detail |

Halaman : 1

**Gambar 5.11** Tampilan Halaman Konfirmasi Pemesanan Member Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

b. Daftar Keranjang

Pada halaman ini bertujuan untuk menampilkan semua daftar pembelanjaan yang belum di konfirmasi.



Shopping Chart

Untuk mengubah jumlah barang, ketik jumlah yang dikehendaki di kolom QTY. Periksalah barang yang telah Anda pesan dengan benar. Lalu, pilihlah jasa pengiriman yang ingin Anda gunakan. Klik tombol "LANGKAH BERIKUTANYA" untuk melanjutkan ke langkah berikutnya, atau silahkan log in dengan cara memasukkan alamat email dan password jika Anda sudah menjadi anggota, untuk mempercepat proses belanja.

| PRODUK | HAPUS | UKURAN | BERAT (kg) | QTY | DISKON (%) | HARGA | SUB TOTAL |
|---|---|---|---|---|---|---|---|
| Baju 21 | HAPUS | L | 0.5 | 2 | 0 | 70.000,00 | 140.000,00 |
| Sepatu 001 Kuat, nyaman, style anak muda | HAPUS | 40 | 1.5 | 1 | 5 | 200.000,00 | 190.000,00 |
| | | | | | | TOTAL BELANJA (2 produk) 2.5 kg | 330.000,00 |

**Jasa Pengiriman** yang akan digunakan :

Pilih Jasa Pengiriman → Pilih Layanan Cargo → Pilih Kota Tujuan 0,00

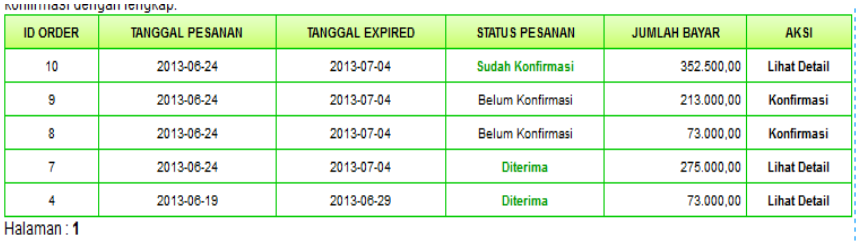
TOTAL BAYAR 330.000,00

← Belanja Lagi Lanjutakan

**Gambar 5.12** Tampilan Halaman Daftar Keranjang Member Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

c. Daftar Konfirmasi Pembayaran

Pada halaman ini member dapat mengetahui data pembayaran yang belum maupun yang sudah terkonfirmasi.



| ID ORDER | TANGGAL PESANAN | TANGGAL EXPIRED | STATUS PESANAN | JUMLAH BAYAR | AKSI |
|---|---|---|---|---|---|
| 10 | 2013-06-24 | 2013-07-04 | Sudah Konfirmasi | 352.500,00 | Lihat Detail |
| 9 | 2013-06-24 | 2013-07-04 | Belum Konfirmasi | 213.000,00 | Konfirmasi |
| 8 | 2013-06-24 | 2013-07-04 | Belum Konfirmasi | 73.000,00 | Konfirmasi |
| 7 | 2013-06-24 | 2013-07-04 | Diterima | 275.000,00 | Lihat Detail |
| 4 | 2013-06-19 | 2013-06-29 | Diterima | 73.000,00 | Lihat Detail |

Halaman : 1

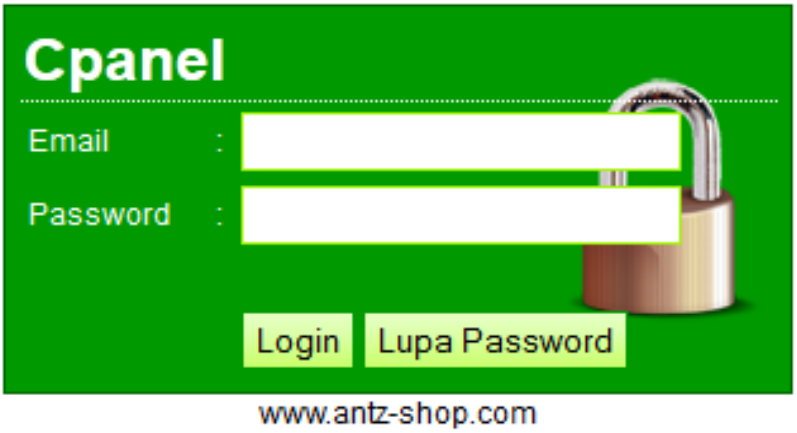
**Gambar 5.13** Tampilan Halaman Konfirmasi Pembayaran Member Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

#### 4.4.7.2 Pengguna Program User Petugas

**1. Tampilan Input Petugas**

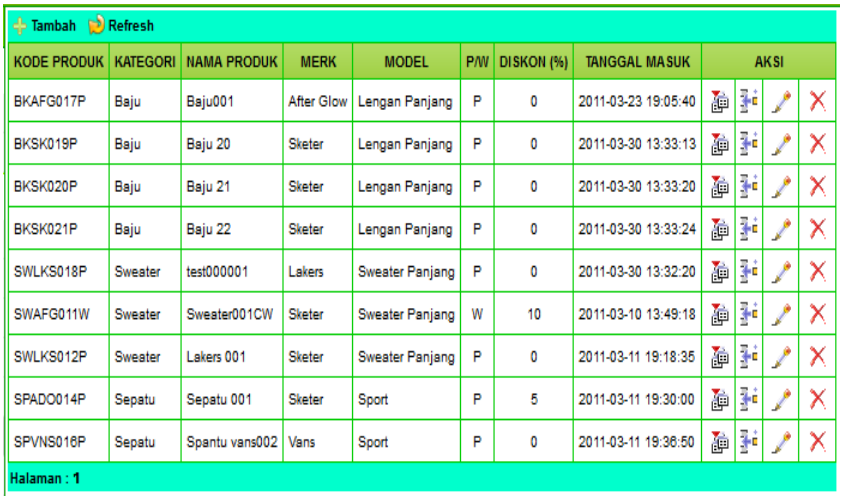
a. Halaman Login Petugas.

Halaman Tampilan Petugas merupakan tahapan sebelum masuk kehalaman petugas, dimana Petugas dapat melakukan perubahan pada content website AntzShop. Jika *username* atau *password* tidak diisi atau salah maka Petugas akan di kembalikan kembali kepada halaman login.




**Gambar 5.14** Tampilan Login Petugas Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

b. Form Input Produk




**Gambar 5.15** Tampilan Form Tambah Produk Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

**2. Tampilan Output petugas**

a. Halaman Kelola Penjualan

Pada Halaman daftar pemesanan,petugas dapat melihat daftar pemesanan produk pada website AntzShop dan dapat merubah status pemesanan.


Refresh

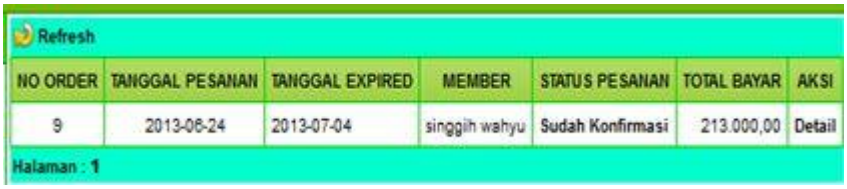
| NO ORDER | TANGGAL PESANAN | TANGGAL EXPIRED | MEMBER | STATUS PESANAN | TOTAL BAYAR | AKSI |
|---|---|---|---|---|---|---|
| 9 | 2013-06-24 | 2013-07-04 | singgih wahyu | Belum Konfirmasi | 213.000,00 | Detail |
| 8 | 2013-06-24 | 2013-07-04 | singgih wahyu | Belum Konfirmasi | 73.000,00 | Detail |

Halaman : 1

**Gambar 5.16** Tampilan Halaman daftar pemesanan produk Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web*.

b. Halaman Kelola Pembayaran

Pada Halaman daftar pembayaran,petugas dapat melihat daftar pembayaran yang telah di transfer oleh member


Refresh

| NO ORDER | TANGGAL PESANAN | TANGGAL EXPIRED | MEMBER | STATUS PESANAN | TOTAL BAYAR | AKSI |
|---|---|---|---|---|---|---|
| 9 | 2013-06-24 | 2013-07-04 | singgih wahyu | Sudah Konfirmasi | 213.000,00 | Detail |

Halaman : 1

**Gambar 5.17** Tampilan Halaman daftar pembayaran Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web*.

c. Halaman Kelola Member

Pada halaman daftar member ini,petugas dapat melihat semua daftar member yang telah terdaftar pada *website* AntzShop



Refresh

| No | EMAIL | NAMA LENGKAP | ALAMAT | KOTA | PHONE | STATUS | INSERT DATE | LAST LOGIN | AKSI |
|---|---|---|---|---|---|---|---|---|---|
| 1 | xxx@gmail.com | xxx xxx | xxx | Jakarta Selatan | 123 | Aktif | 2013-06-18 01:31:48 | 2013-06-18 01:43:56 | |
| 2 | singgihw12@gmail.com | singgih wahyu | sukarajin 2 | Jakarta Pusat | 0223 | Aktif | 2013-06-19 15:15:13 | 2013-06-24 04:51:32 | |
| 3 | singgih@gmail.com | singgih wahyu | sukarajin | Majalengka | 09876 | Aktif | 2013-06-19 17:05:42 | 2013-06-19 17:07:12 | |

Halaman : 1

**Gambar 5.18** Tampilan Halaman daftar member Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web*..

d. Halaman Kelola Produk

Pada Halaman Daftar Produk, Petugas dapat melihat daftar Produk pada website AntzShop, dapat melakukan perubahan dan penghapusan pada setiap Produk.



Tambah Refresh

| KODE PRODUK | KATEGORI | NAMA PRODUK | MERK | MODEL | P/W | DISKON (%) | TANGGAL MASUK | AKSI |
|---|---|---|---|---|---|---|---|---|
| BKAFG017P | Baju | Baju001 | After Glow | Lengan Panjang | P | 0 | 2011-03-23 19:05:40 | |
| BKSK019P | Baju | Baju 20 | Sketer | Lengan Panjang | P | 0 | 2011-03-30 13:33:13 | |
| BKSK020P | Baju | Baju 21 | Sketer | Lengan Panjang | P | 0 | 2011-03-30 13:33:20 | |
| BKSK021P | Baju | Baju 22 | Sketer | Lengan Panjang | P | 0 | 2011-03-30 13:33:24 | |
| SWLKS018P | Sweater | test000001 | Lakers | Sweater Panjang | P | 0 | 2011-03-30 13:32:20 | |
| SWAFG011W | Sweater | Sweater001CW | Sketer | Sweater Panjang | W | 10 | 2011-03-10 13:49:18 | |
| SWLKS012P | Sweater | Lakers 001 | Sketer | Sweater Panjang | P | 0 | 2011-03-11 19:18:35 | |
| SPADO014P | Sepatu | Sepatu 001 | Sketer | Sport | P | 5 | 2011-03-11 19:30:00 | |
| SPVNS016P | Sepatu | Spantu vans002 | Vans | Sport | P | 0 | 2011-03-11 19:36:50 | |

Halaman : 1

**Gambar 5.19** Tampilan Halaman Daftar Produk Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis *Web.*

## 4.5. Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Tujuan dari pengujian adalah untuk menjamin perangkat lunak yang di bangun memilki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Dalam analisis dan perancangan ini, penulis menggunakan 3 (tiga) faktor pengujian yang dilakukan secara internal terhadap *Web* AntzShop pada *module* penjualan *online*.

### 4.5.1 Rencana Pengujian

Pengujian perangkat lunak Sistem Informasi Penjualan Online ini menggunakan metode *black box.* Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat.

**Tabel 5.4** Rencana Pengujian Perancangan Sistem Informasi Penjualan pada Toko AntzShop Berbasis Web.

| Kelas Uji | Butir Uji | Tingkat Pengujian | Jenis Pengujian |
|---|---|---|---|
| Login | Hak akses pengguna program | Modul | *Black box* |
| Informasi Produk | Menampilkan informasi daftar produk | Modul | *Black box* |
| Informasi Konfirmasi | Menampilkan informasi konfirmasi pemesanan produk | Modul | *Black box* |

| | | | |
|---|---|---|---|
| Pengolahan data produk | Penambahan, pengeditan, penyimpanan dan penghapusan data produk | Modul | *Black box* |

### 4.5.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian dibuat untuk mengetahui apakah Sistem Informasi Penjualan pada Toko AntzShop berbasis *web* dapat berjalan dengan baik atau tidak.

#### 4.5.2.1 Kelas Uji Login

Berikut ini adalah tabel pengujian *login Petugas* untuk pengecekkan *user* yang telah terdaftar :

**Tabel 5.5** Pengujian Login Petugas

| **Kasus dan Hasil Uji (Data normal)** | | | |
|---|---|---|---|
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| *User name :* singgih<br>*Password :* 121298 | Dapat Login dan masuk ke halaman utama petugas | Dapat login dan masuk ke halaman utama petugas | [ X ] Diterima<br>[ ] Ditolak |
| *Login* | Dapat masuk ke halaman utama petugas | Tombol Simpan dapat berfungsi sesuai yang diharapkan | [ X ] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data salah)** | | | |
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| *User name* : bebas | Tidak dapat | Petugas tidak dapat | [ ] Diterima |

| | | | |
|---|---|---|---|
| *Password* : bebas | Login dan Menampilkan pesan kesalahan dan masuk lagi kehalaman petugas | login jika salah Username dan Password | [ X ] Ditolak |

#### 4.5.2.2 Kelas Uji Informasi Produk

Berikut adalah tabel pengujian informasi produk :

**Tabel 5.6** Pengujian Informasi Produk

| **Kasus dan Hasil Uji (Data normal)** | | | |
|---|---|---|---|
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| Input data produk | Saat tekan tombol simpan, data dapat tersimpan ke database sesuai yang diharapkan | Tombol input dapat melakukan fungsinya dengan baik | [ X ] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data salah)** | | | |
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| Kotak inputan di isi dengan data yang tidak ada atau data kosong. | Menampilkan kotak pesan kesalahan | Sesuai dengan yang diharapkan | [ ] Diterima<br>[ X ] Ditolak |

### 4.5.2.3 Kelas Uji Informasi Konfirmasi Pembayaran

Berikut adalah tabel pengujian konfirmasi pembayaran:

**Tabel 5.7** Pengujian Informasi Konfirmasi Pembayaran

| **Kasus dan Hasil Uji (Data normal)** | | | |
|---|---|---|---|
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| Pengisian form Konfirmasi pembayaran oleh member. Kemudian menekan tombol kirim | Saat tekan tombol kirim, data dari pemesanan yang telah dilakukan konfirmasi pembayaran tersebut akan ditampilkan pada menu transaksi yang berada pada halaman akun petugas dengan status menunggu proses dilakukanya *acc* oleh petugas | form konfirmasi pembayaran akan hilang dengan sendirinya apabila telah dilakukanya konfirmasi oleh petugas | [ X ] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data salah)** | | | |
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| Tidak melakukan konfirmasi pembayaran. | Data status pemesanan pada menu daftar pesanan yang ada pada halaman akun petugas tidak berubah | Tidak akan ditampilkan pada menu transaksi halaman utama petugas guna dilakukannya proses *acc* oleh petugas | X ] Diterima<br>[ ] Ditolak |

#### 4.5.2.4 Kelas Uji Pengolahan Data Produk

Berikut adalah tabel pengujian pengolahan data produk :

**Tabel 5.8** Pengujian Pengolahan Data Produk

| **Kasus dan Hasil Uji (Data normal)** | | | |
|---|---|---|---|
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| Input data produk | Saat tekan *enter* atau klik tombol simpan, data yang diinputkan masuk ke *database* | Sesuai yang diharapkan | [ X ] Diterima<br>[ ] Ditolak |
| Klik Text Link edit | Akan tampil ke halaman edit data produk, dan ketika klik tombol edit produk maka akan mengedit data produk yang sebelumnya telah diinputkan | Sesuai yang diharapkan | [ X ] Diterima<br>[ ] Ditolak |
| Klik Text Link delete | Akan menghapus data yang akan dihapus | Sesuai yang diharapkan | [ X ] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data salah)** | | | |
| **Data Masuk** | **Yang Diharapkan** | **Pengamatan** | **Kesimpulan** |
| Kotak inputan tidak diisi seperti yang telah ditentukan. | Menampilkan kotak pesan kesalahan. | Sesuai dengan harapan | [ ] Diterima<br>[ X ] Ditolak |

## 4.6. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* di atas, dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.