

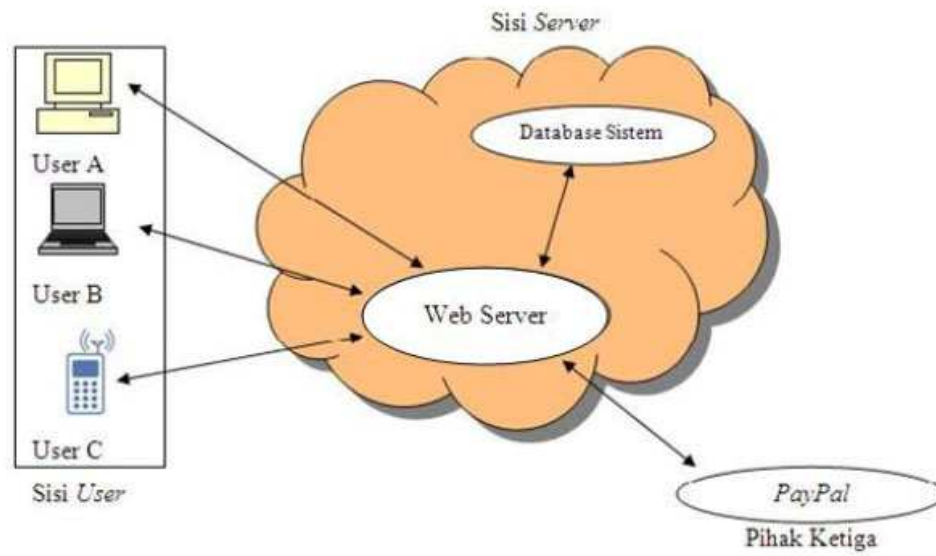
### 3. PERENCANAAN DAN IMPLEMENTASI SISTEM

Implementasi dari skripsi ini berfokus pada pembangunan suatu situs yang terdiri dari dua buah bagian, yaitu *website* dan *mobile site* yang memiliki sebuah inti program yang sama, *database* yang dipakai bersama dan memiliki tampilan yang berbeda. Pada kedua situs tersebut, *database* yang dipakai benar-benar merupakan suatu kesatuan yang tidak bisa terpisahkan. Hal ini dikarenakan agar tidak terjadi kerancuan apabila terdapat transaksi pada salah satu situs. Selain pada bagian *multisite* tersebut, fokus pada skripsi ini adalah bagaimana dapat mewujudkan situs penjualan yang dapat berjalan dengan sesederhana mungkin, namun masih memiliki fungsionalitas yang tinggi. Hal ini dapat terlihat dari jumlah fitur yang dimiliki, meskipun tidak terlalu banyak, tetapi semua memang benar diperlukan untuk kelangsungan daripada toko *online* tersebut.

#### 3.1. Eksplorasi Kebutuhan Fitur Awal Situs

Kebutuhan awal dari halaman situs ini merupakan sebuah situs penjualan yang memiliki fitur yang lengkap namun cukup sederhana untuk dapat dipahami bahkan oleh pengguna yang masih awam sekalipun. Selain itu karena situs penjualan ini akan dapat dibuka baik melalui *browser* maupun melalui *mini browser*, maka situs penjualan ini diharapkan dibuat sesederhana mungkin tanpa mengurangi fungsinya agar tidak memberatkan para penggunanya yang membuka situs ini melalui *mini browser*.

Gambaran dari sistem yang ada adalah calon pembeli atau *user* dapat mengakses halaman *web* maupun *mobile web* ini melalui berbagai macam cara. Seperti pada gambaran dibawah, maka *user A* dan *B* mengakses halaman *web* dengan menggunakan *browser* biasa melalui PC ataupun *laptop*. Koneksi ini menggunakan koneksi *internet* seperti *wireless internet*, *hotspot*, maupun *LAN*. Pada *user C* terlihat bahwa *user C* menggunakan *handphone* yang akan mengakses halaman *mobile web* menggunakan *mini browser* yang terdapat pada *handphone* tersebut.



Gambar 3.1. Gambaran Sistem yang Dipakai

Akan terdapat tiga macam *user* pada halaman *e-commerce* ini, antara lain *guest*, *member* dan *admin*. Fitur-fitur yang didapat antara lain :

Fitur dasar yang dimiliki *guest*:

- Melihat aneka kue yang dijual beserta harganya
- Melakukan transaksi secara online
- Mengecek apakah transaksi berhasil dilakukan atau tidak.
- Mendapat konfirmasi berupa email yang akan dikirimkan ke alamat email yang diberikan oleh *guest*.

Fitur yang dimiliki oleh *member* adalah semua fitur dasar yang dimiliki *guest*, dan juga beberapa fitur tambahan, antara lain :

- Terdapat halaman *login*
- Mengecek selama ini pembelian yang telah dilakukan
- Mengecek pengiriman (*tracking*)
- Melihat apakah kue yang akan dibeli *ready stock* ataukah masih harus menunggu.

Fitur yang dimiliki oleh *admin* adalah semua fitur dasar yang dimiliki *guest* dan *member*, dan juga beberapa fitur tambahan, antara lain :

- Melakukan masukan dan *update* katalog serta harga barang
- Melihat rekapitulasi penjualan hari ini dan bulanan
- Melihat status pembayaran transaksi
- Memberikan informasi status pengiriman barang
- *User management*

Untuk melakukan transaksi pembelian kue, tidak dibutuhkan registrasi terlebih dahulu. Bagi para tamu, atau *guest* dapat melakukan transaksi pembelian secara langsung tanpa melakukan registrasi maupun *login* terlebih dahulu. Kelebihan dari registrasi dan menjadi member adalah mendapatkan potongan harga pada beberapa jenis barang yang dijual. Selain itu, untuk member, pelacakan pengiriman dan transaksi selain dikirimkan melalui *email* ke alamat *email* member, juga terdapat transaksi pada halaman *web* tersebut pada bagian *member area*. Untuk perbedaan fitur antara ketiga jenis *login* diatas, untuk *member* dan *guest* diperuntukkan untuk calon pembeli. Sedangkan *login admin* diperuntukkan bagi penjual.

Kebutuhan awal dari situs ini antara lain adalah :

- Pengunjung situs dapat melihat dan melakukan pemesanan terhadap barang yang dijual pada situs tersebut.
- Pembayaran dapat dilakukan secara *online* melalui *paypal*.
- Pengunjung situs yang melakukan pembelian mendapatkan pemberitahuan melalui *email* yang dikirimkan ke alamat *email* mereka.
- Pada bagian administrasi juga mendapatkan *email notification* apabila terjadi pembelian, maupun pembayaran yang dilakukan oleh seorang pembeli.
- Pada bagian halaman administrasi, seorang *admin* dapat melakukan *update* jumlah barang yang tersedia, maupun perubahan harga dan status pembelian yang dilakukan oleh seorang anggota dari situs tersebut.

### 3.2. Desain Awal *Template* Situs Penjualan

Karena situs penjualan ini memiliki dua buah bagian, maka desain awal dari situs ini juga terdiri dari dua macam desain. Desain awal untuk halaman situs yang diakses melalui *browser* biasa, mengambil *template* dari Drupal yang telah tersedia secara luas dan gratis, sedangkan untuk desain dari halaman situs yang dibuka melalui *mini browser* digunakan *template mobile* yang memang disiapkan untuk pengguna *mobile* yang akan membuka situs tersebut dari peralatan genggam mereka.

### 3.3. Desain Awal Program Situs Penjualan

Desain awal program situs penjualan ini memiliki dua buah situs yang berbeda dengan isi yang sama. Perbedaannya terdapat pada tampilan dan cara aksesnya yang berbeda. *Database* yang dimiliki oleh kedua situs tersebut merupakan *database* tunggal yang sinkron antara satu dengan yang lainnya. Jadi apabila terjadi suatu transaksi pada salah satu situs, maka secara otomatis, kedua situs akan memiliki laporan yang sama. Untuk dapat melakukan hal ini, maka dimanfaatkanlah *multisites* yang memungkinkan drupal untuk memiliki banyak situs dengan hanya satu inti dan satu *database* yang dapat diakses secara bersamaan.

Perubahan *database* ini akan dilakukan pada bagian *setting.php* dibawah folder *sites*. Untuk tiap domain akan memiliki sebuah folder yang sesuai dengan nama domain tersebut. Maka untuk *setting.php* akan ditempatkan di bawah folder tersebut.

#### 3.3.1. Desain *Database* yang Harus Dipisahkan

Dalam mencapai tujuan diatas, maka tabel *database* inti dari drupal yang harus dipisahkan antara lain adalah :

- *Cache* dari situs tersebut. Yang termasuk di dalamnya antara lain adalah :
  - *Cache*
  - *Cache\_filter*
  - *Cache\_menu*

- *Cache\_page*
  - *Variable*
- *Site tracking*, yang termasuk di dalamnya adalah :
  - *Acceslog*
  - *Client*
  - *Client\_system*
  - *Flood*
  - *Watchdog*
- *Theme layout*. Pada bagian ini diatur tampilan dari situs tersebut. Yang termasuk di dalamnya antara lain :
  - *Blocks*
  - *Blocks\_roles*
  - *Boxes*
  - *Menu*
  - *System*
- Selain bagian-bagian tersebut yang harus dipisahkan adalah *sequences*.

Apabila bagian-bagian tersebut tidak dipisahkan, maka yang terjadi adalah duplikasi dari situs tersebut tanpa adanya perbedaan tampilan dari kedua situs tersebut.

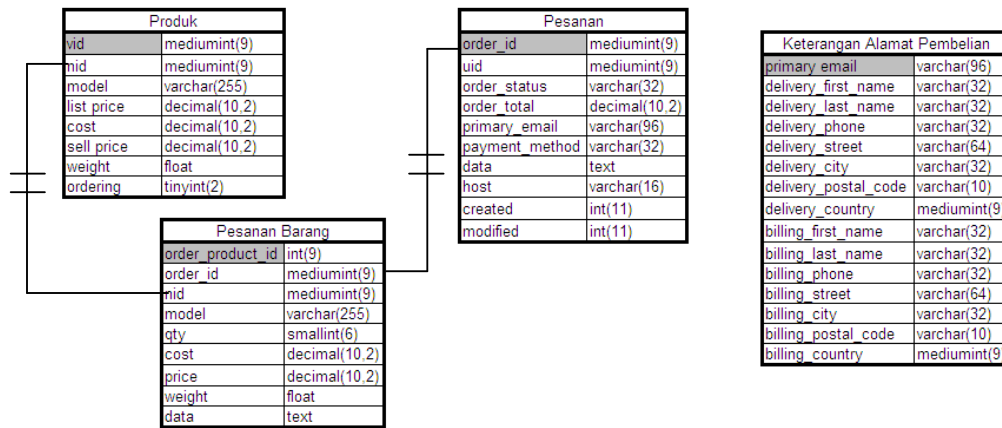
### **3.3.2. Database yang Dipakai Bersama (*Shared Database*)**

Selain tabel-tabel yang disebutkan diatas, maka tabel-tabel lain dari *database* tersebut sebaiknya dibuat *shared database*. Hal ini akan sangat membantu untuk proses *update* situs tersebut, baik dalam administrasi produk maupun dalam administrasi situs dan penambahan anggota situs tersebut.

### **3.3.3. Database Pemesanan Produk**

Pada desain pemesanan produk ada tiga hal yang utama yaitu untuk bagian produk, pesanan dan pesanan produk.

ERD dari *database* tersebut adalah :



Gambar 3.2. ERD *database*

### 3.3.3.1. Desain *Database* Pesanan Pembelian

Untuk dapat melakukan pemesanan, seorang pengunjung situs tidak perlu melakukan registrasi terlebih dahulu, tetapi pada akhirnya apabila pengunjung tersebut memutuskan untuk melakukan pembelian pada situs ini, maka pengunjung ini akan diminta memberikan informasi tentang data-data pribadi dirinya. Data-data tersebut, selain dibutuhkan untuk melakukan pengiriman, juga menjadi bagian dari arsip situs akan pembeli yang pernah melakukan transaksi.

Jadi apabila seorang pengunjung melakukan transaksi, secara otomatis ketika ia mengisi data pengiriman dan data pembayaran, maka pengunjung tersebut akan langsung terdaftar sebagai seorang anggota dari situs penjualan ini. Ketika sudah menjadi anggota, maka dia akan dapat melihat status dari transaksi yang pernah dilakukannya, serta status dari transaksi tersebut. Status keanggotaan ini juga memudahkan untuk administrasi dan keperluan arsip dari situs penjualan ini. Untuk itu, maka data yang akan diperlukan seperti pada tabel 3.1.

Tabel 3.1. Tabel Data Pesanan

Pesanan	
order_id	mediumint(9)
uid	mediumint(9)
order_status	varchar(32)
order_total	decimal(10,2)
primary_email	varchar(96)
payment_method	varchar(32)
data	text
host	varchar(16)
created	int(11)
modified	int(11)

Tabel 3.2. Tabel Keterangan Alamat Pembelian

Keterangan Alamat Pembelian	
primary_email	varchar(96)
delivery_first_name	varchar(32)
delivery_last_name	varchar(32)
delivery_phone	varchar(32)
delivery_street	varchar(64)
delivery_city	varchar(32)
delivery_postal_code	varchar(10)
delivery_country	mediumint(9)
billing_first_name	varchar(32)
billing_last_name	varchar(32)
billing_phone	varchar(32)
billing_street	varchar(64)
billing_city	varchar(32)
billing_postal_code	varchar(10)
billing_country	mediumint(9)

Semua data diatas harus diisi, data tersebut nantinya akan terduplikasi menjadi dua bagian, yaitu data pengiriman (*delivery*) dan data penagihan (*billing*). Pembeli nantinya dapat memilih apabila data penagihan sama dengan data pengiriman, maka akan ada pilihan yang memungkinkan pembeli hanya mengisi satu kali saja dari data tersebut.

Untuk menjadi anggota, maka data yang diminta lebih sederhana daripada data tersebut. Hal ini dikarenakan pada awalnya diasumsikan bahwa seorang anggota belum tentu telah melakukan transaksi pembelian. Jadi apabila seorang pengunjung memilih untuk melakukan registrasi sebelum melakukan transaksi,

maka data yang diminta hanya berupa data sederhana yang terdiri dari nama dan alamat email dari pengunjung tersebut. Tetapi nantinya pada saat melakukan transaksi, tetap akan diminta untuk mengisi data pengiriman dan data penagihan tersebut. Pada anggota yang telah mengisi data tersebut, maka apabila melakukan transaksi lagi, akan ditanya apakah data yang pernah diisi sama dengan alamat pengiriman yang akan dilakukan. Apabila sama maka tidak perlu melakukan pengisian data lagi.

### **3.3.3.2. Desain Pemesanan Produk**

Untuk melakukan transaksi pemesanan maupun untuk melihat produk, tidak dibutuhkan registrasi sebagai anggota terlebih dahulu. Hal ini memungkinkan karena nantinya apabila terjadi transaksi, maka akan diminta data dari pembeli tersebut. Hal ini memudahkan untuk melakukan transaksi, sehingga dapat menarik lebih banyak minat untuk melakukan pembelian. Untuk pemesanan produk, selain ada *cart* atau keranjang belanja yang menampung isi dari pesanan seorang calon pembeli, maka pembeli nantinya apabila akan melakukan pembelian akan diminta untuk melakukan konfirmasi dan pengisian data terlebih dahulu.

Selain pengisian data dan konfirmasi, maka pembeli nantinya akan diminta untuk melakukan pemilihan cara dan biaya pengiriman yang akan dilakukan. Selain itu, pembayaran juga dapat dilakukan dengan tiga cara yang nantinya akan dipilih oleh pembeli.

Struktur *database* yang akan dipakai untuk pemesanan ini terlihat seperti pada tabel 3.3.

Tabel 3.3. Tabel Data Pesanan Barang

Pesanan Barang	
order_product_id	int(9)
order_id	mediumint(9)
nid	mediumint(9)
model	varchar(255)
qty	smallint(6)
cost	decimal(10,2)
price	decimal(10,2)
weight	float
data	text

### 3.3.3.3. Desain Database Produk

Pada desain *database* produk ini, yang dipikirkan adalah adanya kategori yang nantinya dapat memisah-misahkan antara satu buah produk dengan produk yang lain. Kategori ini nantinya dapat memiliki banyak produk, dan satu buah produk tidak harus masuk kedalam satu buah kategori saja, melainkan dapat masuk ke dalam banyak kategori. Selain itu, yang harus dipikirkan adalah agar dapat menghitung biaya pengiriman, maka sebuah barang akan dianggap sebagai sebuah paket. Hal ini dipakai untuk memudahkan pengiriman saja.

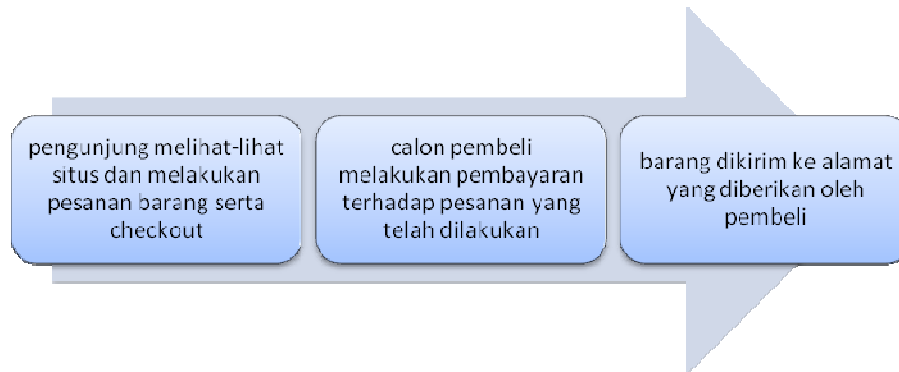
Desain *database* akan dibuat seperti ini :

Tabel 3.4. Tabel Database Produk

Produk	
Vid	mediumint(9)
Nid	mediumint(9)
Model	varchar(255)
list price	decimal(10,2)
Cost	decimal(10,2)
sell price	decimal(10,2)
weight	float
ordering	tinyint(2)

### 3.4. Desain Awal Cara Kerja Sistem

Seperti pada situs belanja *online* lainnya, maka blok diagram dari sistem yang ada adalah seperti pada gambar 3.3.

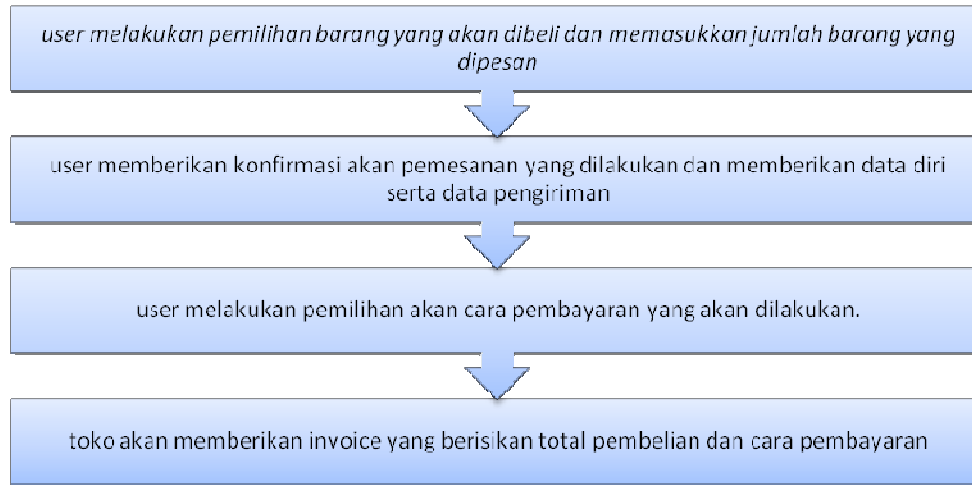


Gambar 3.3. Blok diagram sistem secara keseluruhan

Pada desain awal sistem ini maka pengunjung dari situs tidak harus melakukan registrasi dan *login* terlebih dahulu untuk melakukan transaksi. Setelah melakukan pemesanan dan melakukan konfirmasi, langkah selanjutnya yang dilakukan oleh *user* adalah memberikan identitas dirinya serta memilih metode pembayaran yang akan dilakukan. Pemberian identitas ini dimaksudkan untuk kepentingan *verifikasi* dari transaksi yang akan dilakukan, selain itu pemberian identitas ini akan memberikan informasi kemanakah barang yang telah dibeli akan dikirimkan.

Setelah itu *user* diarahkan untuk melakukan pembayaran dan konfirmasi akan pembayaran tersebut. Setelah konfirmasi telah diterima oleh pemilik situs, maka barang yang telah dipesan akan dikirimkan ke tempat pemesan.

### 3.5. Desain Awal Perencanaan Pemesanan

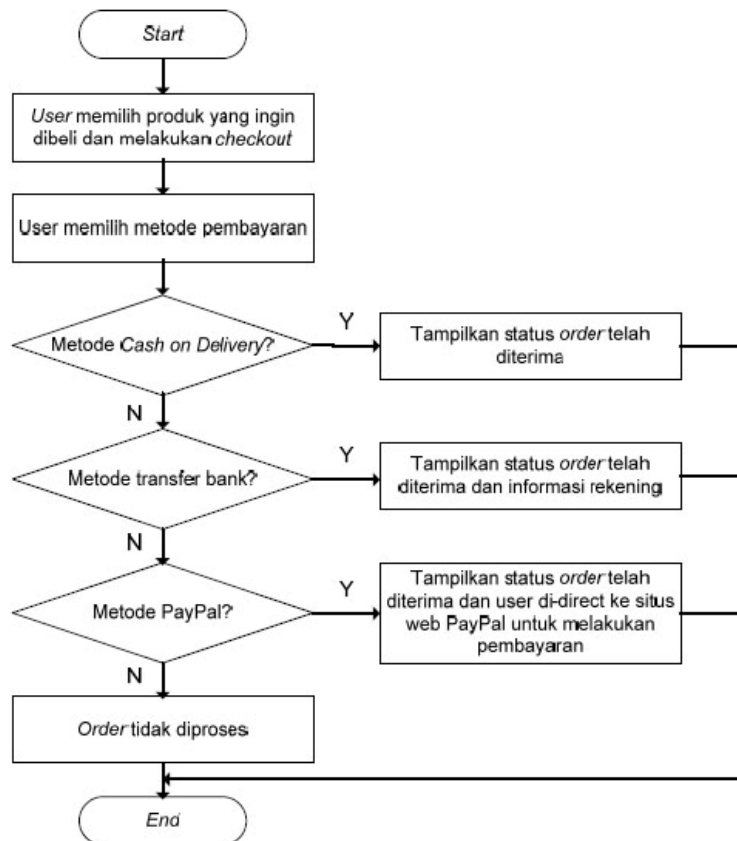


Gambar 3.4. Blok diagram pemesanan barang

Pada bagian pemesanan ini, maka langkah-langkah yang akan dilakukan oleh pengunjung situs adalah :

1. Pengunjung melihat-lihat barang yang dijual pada situs tersebut. Apabila pengunjung situs tersebut tertarik dengan salah satu barang pada situs tersebut maka pengunjung dapat melakukan pemesanan barang tersebut dan memasukkan jumlah dari barang yang dipesan.
2. Setelah itu, calon pembeli akan diminta untuk melakukan konfirmasi dan memberikan data-data untuk keperluan pengiriman dan keperluan konfirmasi serta pengiriman *invoice*.
3. Setelah data yang diberikan telah lengkap, maka calon pembeli akan diminta untuk melakukan pemilihan cara pembayaran yang akan dilakukan untuk membayar barang tersebut.
4. Setelah semua data dan cara pembayaran telah dipilih oleh calon pembeli, maka dari pihak toko *online* akan menerbitkan *invoice* yang akan ditampilkan di situs tersebut dan dikirimkan kepada *email* calon pembeli tersebut. *Invoice* tersebut akan berisi rekapitulasi dari pesanan dan cara pembayaran yang dipilih oleh calon pembeli.

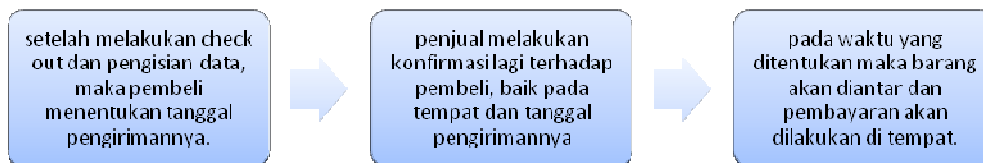
Untuk mengisi data diri dan alamat pengiriman, calon pembeli tidak diharuskan untuk melakukan registrasi terlebih dahulu, namun apabila calon pembeli sebelumnya belum pernah melakukan transaksi dan tidak memiliki akun pada situs ini, setelah melakukan pemesanan dan mengisi data diri serta alamat pengiriman, maka secara otomatis calon pembeli tersebut memiliki akun pada situs ini. Untuk calon pembeli yang telah memiliki akun pada situs penjualan ini, juga tidak diharuskan untuk melakukan *login* terlebih dahulu, tetapi ketika melakukan pengisian data, maka apabila calon pembeli telah memiliki akun pada situs maka akan diberi tawaran untuk melakukan *login* sehingga untuk mengisi data pengiriman tidak perlu mengisi lagi apabila alamat tersebut sama dengan alamat yang lalu. Namun apabila terdapat perbedaan alamat, maka calon pembeli tetap dapat memberikan alamat yang berbeda tersebut dan tetap akan disimpan untuk keperluan transaksi lagi.



Gambar 3.5. Flowchart pemilihan cara pembayaran

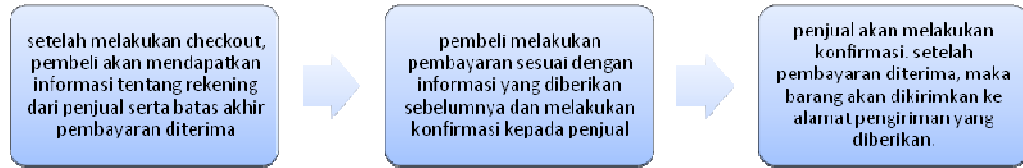
Pada gambar 3.5 merupakan flowchart dari cara pembayaran yang dapat dilakukan pada situs ini. Situs ini memiliki tiga macam pilihan cara pembayaran, yaitu secara transfer antar bank, COD, maupun secara *online* langsung dengan memanfaatkan jasa layanan *paypal*. Layanan *paypal* merupakan layanan dari pihak ketiga dimana nantinya pembeli akan diarahkan ke halaman pembayaran milik situs *paypal* untuk melakukan pembayaran, sehingga nantinya pembeli tidak harus berbagi informasi tentang kartu kredit ataupun rekening yang dimilikinya kepada pihak penjual. Selain itu, dengan memanfaatkan layanan dari pihak ketiga, yaitu *paypal*, pengguna yang tidak memiliki akun *paypal* akan tetap dapat melakukan pembayaran dengan menggunakan kartu kredit milik mereka yang nantinya akan masuk ke rekening *paypal* milik penjual.

Apabila calon pembeli melakukan pemilihan untuk membayar dengan menggunakan metode *cash on delivery*, maka blok diagramnya adalah tergambar seperti pada gambar 3.6.



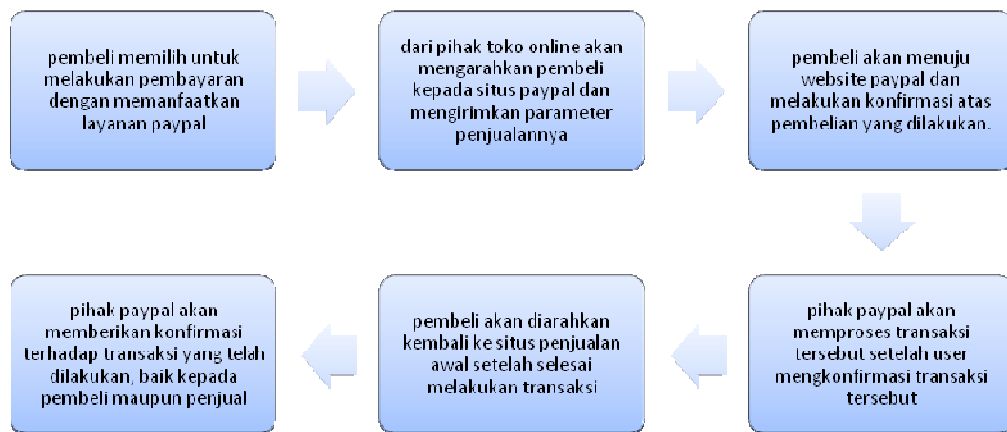
Gambar 3.6. Blok diagram *cash on delivery*

Apabila calon pembeli melakukan pemilihan untuk membayar dengan mentransfer ke rekening penjual, maka blok diagramnya menjadi seperti pada gambar 3.7. Blok diagram pada gambar 3.7. menjelaskan secara langkah demi langkah dari pembayaran dengan menggunakan jasa transfer antar bank.



Gambar 3.7. Blok diagram transfer antar bank

Perbedaan akan terjadi apabila pembeli melakukan pemilihan untuk melakukan pembayaran melalui website *paypal*. Hal ini dikarenakan pembeli akan diarahkan menuju situs *paypal* untuk melakukan *login* dan mentransfer sejumlah uang kepada penjual melalui situs tersebut. Setelah transaksi berhasil dilakukan, maka pihak *paypal* akan memberikan konfirmasi terhadap kedua belah pihak sehingga pesanan tersebut dapat diproses pengirimannya. Oleh karena itu, maka blok diagramnya akan menjadi seperti pada gambar 3.8.



Gambar 3.8. Blok diagram pembayaran melalui *paypal*

*Gateway* pembayaran melalui *paypal* ini memanfaatkan modul tambahan dari Drupal yang bernama *uc\_payment*. Parameter yang harus diatur dari halaman administrasi ada beberapa, antara lain adalah :

- Alamat *email* dari pemilik toko dimana dana tersebut akan ditransfer.

- Pemilihan server *paypal*, apakah melalui *sandbox* atau secara *real* melalui [www.paypal.com](http://www.paypal.com)

### 3.6. Implementasi *Multisites* Pada Situs

Untuk implementasi dari situs ini, dimanfaatkanlah CMS drupal yang mendukung keperluan daripada skripsi ini yang memerlukan agar kedua buah situs memiliki sebuah inti dan sebuah *database* yang berbagi dengan tampilan yang berbeda. Oleh karena itu, maka untuk pengimplementasinya akan dilakukan perubahan pada bagian inti drupal, yaitu pada *folder sites*.

Struktur awal dari *folder sites* adalah sebagai berikut :

- */drupal/sites/all* yang memiliki isi */modules* dan */themes* yang digunakan secara keseluruhan oleh seluruh isi situs tersebut.
- */drupal/sites/default* yang memiliki isi *folder /files* dan *file settings.php* yang merupakan setting parameter dari situs tersebut.

Untuk dapat mengakomodir kebutuhan untuk membuat *multisites* pada situs ini, maka dibawah *folder sites* akan dibuat *folder* baru sejumlah situs yang akan dibuat. *Folder-folder* tersebut akan diberi nama sesuai dengan situs tersebut. Isi daripada *folder* tersebut adalah *folder files*, *modules*, *themes*, *tmp* serta *settings.php*.

*Folder files*, *modules*, dan *themes* juga ada pada *core* drupal, hanya saja terdapat perbedaan fungsi daripada ketika *folder* tersebut ketika ditempatkan di bawah *folder sites* ataupun pada *core* drupal. Apabila ketiga *folder* tersebut ditempatkan dibawah *folder sites*, maka isi dari ketiga *folder* tersebut hanya akan digunakan oleh situs yang menaunginya. Namun ketika ketiga *folder* tersebut berada pada *core* atau inti dari drupal, maka isi dari ketiga *folder* tersebut akan digunakan bersama oleh situs-situs yang berada dibawah naungan *core* tersebut.

Selain pada susunan *folder* tersebut, ada juga pada *file .htaccess* dan *settings.php* yang harus diubah isinya. Pada *file .htaccess*, akan ditambahkan kode :

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ /sites/{HTTP_HOST}/files/$1 [L]
```

Namun apabila penambahan pada *file .htaccess* tersebut tidak dapat berjalan dengan lancar, maka pada *settings.php* harus ditambahkan kode :

```
<?php
$conf['file_directory_temp'] = 'sites/'
.substr(strrchr(dirname(__FILE__), '/'), 1) . 'tmp';
$conf['file_directory_path'] = 'sites/'
.substr(strrchr(dirname(__FILE__), '/'), 1) . '/files';
?>
```

Selain instalasi seperti pada kode diatas, *multisites* dapat dicapai dengan menggunakan modul tambahan dari drupal yang bernama *multidomain*. Hanya saja penggunaan *multidomain* ini nantinya akan memerlukan *patch* terhadap *core* dari drupal tersebut. Pengerjaan *patch* harus dilakukan di level SSH, dan tidak semua orang dapat menguasai hal seperti itu.

Selain menggunakan *plain coding* seperti diatas, atau menggunakan modul dari drupal, pembuatan *multisites* juga dapat dilakukan dengan memanfaatkan fasilitas *parked domain* yang tersedia di tempat kita melakukan hostingan tersebut. Penggunaan fasilitas ini memudahkan para instalatir drupal yang kewalahan dalam mengatasi instalasi *multisites* yang agak rumit.

Contoh pemakaian dari *code program* diatas dapat dilihat pada lampiran 3 sampai dengan lampiran 5. Pada halaman tersebut adalah contoh dari kode *.htaccess* dan halaman *settings.php* dari *website* dan *mobile web* situs ini.