

## WEB MULTI E-COMMERCE BERBASIS FRAMEWORK CODEIGNITER

D Tri Octafian  
STMIK PalComTech

### *Abstract*

*Product sale uses online media (e-commerce website) is very effective, because the scope is very wide, and transaction can be done anytime and anywhere. Not all companies or sellers have an e-commerce website, because not all people can make an e-commerce website, very high production cost and taking a long time. The goal of this research is building a website that can be used many users, so the users can have an e-commerce website fast and free. Development method that used is prototyping method with Object Oriented Programming (OOP) concept that bases on MVC (Model View Controller) Model. This research produces an e-commerce website that can be together had and used by user for offering products that will be sold.*

**Keywords:** *e-commerce, object oriented, MVC*

### PENDAHULUAN

Teknologi mampu memperluas transaksi penjualan dan pembelian produk secara tradisional menuju transaksi online, yang kita sebut dengan teknologi e-commerce. E-commerce adalah penerapan teknologi yang mendukung transaksi penjualan dan pembelian dilakukan melalui media elektronik yang diimplementasikan dengan sebuah website. E-commerce dapat menjadi fasilitas pendukung bagi penjual sebagai solusi alternatif untuk konsumen melakukan transaksi pembelian produk. Banyak penjual membutuhkan website e-commerce untuk memberikan layanan lebih secara online kepada konsumen mereka dalam meningkatkan kepuasan konsumen.

Tidak semua penjual bisa membangun sebuah website e-commerce, karena untuk membangun sebuah website e-commerce membutuhkan pengetahuan khusus, jika menggunakan pihak ketiga yang menyediakan jasa pembuatan website e-commerce atau mengangkat pegawai sebagai tenaga ahli untuk membangun website e-commerce akan memakan biaya yang cukup mahal dan waktu yang lama.

Berdasarkan kondisi diatas, penelitian ini bertujuan untuk membangun sebuah website *e-commerce* yang mampu digunakan oleh banyak pengguna dalam menyajikan layanan penjualan dan pembelian secara online, sehingga website *e-commerce* ini menjadi solusi alternatif bagi penjual untuk memiliki sebuah website *e-commerce* sendiri.

Penelitian Terdahulu:

Wirdasari melakukan penelitian penerapan website e-commerce dalam menunjang proses bisnis, dan Nugrahani melakukan penelitian untuk mengetahui pengaruh *e-commerce* pada pemasaran produk usaha kecil dan menengah.

## LANDASAN TEORI

### E-Commerce

Menurut Triton (2006), E-commerce adalah membeli atau menjual secara elektronik, kegiatan ini dilakukan pada jaringan internet. Ecommerce adalah satu dari sekian nama yang digunakan orang untuk maksud yang sama, nama-nama lain yang digunakan orang untuk menyebut ecommerce antara lain adalah internet commerce, sering disingkat dengan sebutan ICom. Pengertian ecommerce dapat didefinisikan sebagai perdagangan elektronik dimana bentuk transaksi perdagangan baik membeli maupun menjual dilakukan melalui elektronik pada jaringan internet yang beroperasi selama 24 jam.

### Framework

Menurut Basuki (2010) bahwa *framework* dapat diartikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi untuh tanpa harus membuat semua kodenya dari awal. Saat ini ada banyak *framework* PHP, diantaranya: Zend, Cake PHP, Trax, Symfony, Codeigniter dan sebagainya. Tentu saja, setiap *framework* memiliki kelebihan dan kekurangannya masing-masing. Keuntungan yang dapat diperoleh dari penggunaan *framemork* adalah:

1. Waktu pembuatan aplikasi *website* jauh lebih singkat.
2. Kode aplikasi *website* menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok, detailnya adalah kode dari *framework*.
3. *Website* menjadi lebih mudah diperbaiki, karena tidak perlu fokus ke semua komponen kode *website*, terutama kode sistem *framework*.
4. Tidak perlu lagi membuat kode penunjang aplikasi *website* seperti koneksi *database*, validasi *form*, GUI, dan keamanan.
5. Pikiran pengembang menjadi lebih terfokus ke kode alur permasalahan *website*, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi *website* tersebut.
6. Jika dikerjakan *team work*, maka akan lebih terarah karena sistem *framework*, mengharuskan adanya keteraturan peletakan kode. Seperti bagian pengambilan *database* terpisah dengan bagian pengaturan tampilan untuk pengunjung.

### Codeigniter

Menurut Basuki (2010) bahwa Codeigniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi website berbasis PHP dibandingkan jika menulis semua kode program dari awal. Codeigniter menyediakan banyak *library* untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi berbasis web. Selain itu, struktur dan susunan logis dari Codeigniter membuat aplikasi yang dibuat menjadi semakin teratur dan rapi. Dengan demikian *developer* dapat fokus pada fitur-fitur apa yang dibutuhkan oleh aplikasi dengan membuat kode program seminimal mungkin. Codeigniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. (<http://ellislab.com>), sebuah perusahaan yang memproduksi sebuah CMS (*Content Management System*) yang cukup handal, yaitu ExpressionEngine (<http://www.expressionengine.com>). Saat ini, Codeigniter dikembangkan dan dimaintain oleh ExpressionEngine *Development Team*.

Beberapa keuntungan menggunakan Codeigniter, diantaranya:

1. Gratis  
Codeigniter berlisensi dibawah Apache/ BSD *opensource*, sehingga penggunaannya secara bebas.
2. Ditulis menggunakan PHP 4  
Meskipun Codeigniter dapat berjalan pada PHP 5, namun sampai saat ini kode program Codeigniter masih dibuat dengan menggunakan PHP 4. Hal ini dilakukan agar Codeigniter dapat tersebar lebih luas di komunitas PHP. Karena hingga saat ini, sebagian besar *web hosting* masih menggunakan PHP 4. Jika Codeigniter dibuat dengan PHP 5, tentu saja hasilnya juga akan jauh lebih canggih, karena bisa memanfaatkan teknologi PHP 5 yang saat ini masih belum dapat dilakukan oleh PHP 4, misalnya untuk menerapkan konsep OOP *Multiple Inheritance*.
3. Berukuran kecil  
Ukuran Codeigniter yang kecil merupakan keunggulan tersendiri. Dibandingkan *framework* lain yang berukuran besar, serta membutuhkan *resource* yang besar pula untuk berjalan. Pada Codeigniter, bisa diatur agar sistem *autoload library* yang dibutuhkan saja, sehingga sistem dapat berjalan ringan dan cepat.
4. Menggunakan konsep MVC  
Codeigniter menggunakan konsep MVC (*Model View Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*.
5. URL yang sederhana  
Secara *default*, URL yang dihasilkan Codeigniter sangat bersih (*clean*) dan *Search Engine Friendly* (SEF).
6. Memiliki paket *library* yang lengkap  
Codeigniter memiliki *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*, misalnya mengakses *database*, mengirim *email*, memvalidasi *form*, menangani *session* dan sebagainya.
7. *Extensible*  
Sistem dapat dikembangkan dengan mudah dengan menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.
8. Tidak memerlukan *template engine*  
Meskipun Codeigniter dilengkapi dengan *templateparses* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan untuk menggunakannya. Penggunaan *template engine* dapat mengurangi *performance* dari sistem.
9. Dokumentasi lengkap dan jelas  
Dari sekian banyak *framework*, Codeigniter adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas. Tim pengembang Codeigniter berkomitmen bahwa dokumentasi juga sama pentingnya dengan kode program Codeigniter itu sendiri. *Source code* Codeigniter juga dilengkapi komentar didalamnya, sehingga memperjelas fungsi sebuah kode program.
10. Komunitas  
Komunitas pengguna Codeigniter saat ini berkembang pesat, dan dapat berpartisipasi di <http://codeigniter.com/forums/>.  
Beberapa hal yang harus diketahui dari Codeigniter:

a. Kebutuhan Sistem

Untuk menjalankan Codeigniter diperlukan *server* yang menjalankan PHP versi 4.3.2 atau yang lebih tinggi. Codeigniter dapat mendukung RDBMS MySQL (4.1+), MySQLi, Ms. SQL Server, Postgres, Oracle, SQLite, dan ODBC.

b. Fitur Codeigniter

Berikut ini adalah fitur-fitur yang didukung oleh Codeigniter:

*Model View Controller based, PHP 4 compatible, Extremely light weight., Full featured database classes with support for several platforms, Active record database support, Form and data validation, Security and XSS filtering, Session management, Email sending class, Support attachment, HTML/ Text email, multiple protocols (sendmail, SMTP, dan mail), Image manipulation library (cropping, resizing, rotating), Support GD, ImageMagick dan NetPBM, File uploading class, FTP class, Localization, Pagination, Data encryption, Benchmarking, Full page caching, Error logging, Application profiling, Scaffolding, Calendar class, User agent class, Zip encoding class, Template engine class, Trackback class, XML-RPC library, Unit testing class, Search engine friendly URLs, Flexible URI routing, Support for hooks, class extensions, dan plugins, Large library of helper functions.*

### Teknik Pengembangan *Prototyping*

Menurut Pressman (2002) metode *prototyping* sering digunakan pada dunia nyata. Karena metode ini secara keseluruhan akan mengacu kepada kepuasan pengguna. Bisa dikatakan bahwa metode ini merupakan metode *waterfall* yang dilakukan secara berulang-ulang.

#### Tahapan Metode *Prototyping*

1. Pemilihan Fungsi. Mengacu pada pemilihan fungsi yang harus ditampilkan oleh *prototyping*. Pemilihan harus selalu dilakukan berdasarkan pada tugas-tugas yang relevan yang sesuai dengan contoh kasus yang akan dipergunakan.
2. Penyusunan Sistem Informasi. Bertujuan memenuhi permintaan kebutuhan akan tersedianya *prototype*.
3. Evaluasi.
4. Penggunaan selanjutnya.

#### Jenis Jenis *prototyping*

1. *Feasibility prototyping*. Digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk sistem informasi yang akan disusun.
2. *Requirement prototyping*. Digunakan untuk mengetahui kebutuhan aktivitas bisnis pengguna. Misalnya dalam sebuah perusahaan terdapat pengguna direktur, manajer, dan karyawan. Maka penggunaan sistem dapat dibedakan berdasarkan pengguna tersebut sesuai dengan kebutuhannya.
3. *Design prototyping*. Digunakan untuk mendorong perancangan sistem informasi yang akan digunakan.
4. *Implementation prototyping*. Merupakan lanjutan dari rancangan *prototype*, *prototype* ini langsung disusun sebagai suatu sistem informasi yang akan digunakan.

#### Keunggulan metode *prototyping*

1. Adanya komunikasi baik antara pengembang dengan pelanggan.
2. Pengembang dapat bekerja lebih baik untuk memenuhi kebutuhan pelanggan.

3. Pelanggan berperan aktif dalam pengembangan sistem.
  4. Menghemat waktu dalam pengembangannya.
    5. Penerapan lebih mudah karena pemakai akan mengetahui apa yang diharapkan.
- Kelemahan metode *prototyping*
1. Kualitas sistem kurang baik karena hanya mengedepankan aspek kenyamanan pengguna.
  2. Pengembang kadang-kadang menggunakan implementasi yang sembarangan.
  3. Tidak mencerminkan proses perancangan yang baik.

## UML

Menurut Widodo dkk (2011), UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti, bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

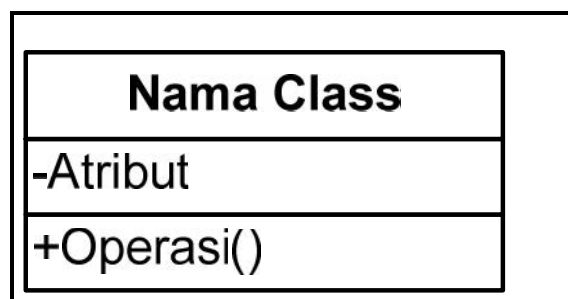
1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Jenis-jenis diagram yang dimiliki oleh UML adalah sebagai berikut:

1. Diagram kelas (class diagram)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* (proses perubahan model menjadi kode program) atau *reverse engineering* (proses perubahan kode program menjadi model) memanfaatkan diagram ini.

Kelas dinyatakan dalam kotak yang terbagi menjadi beberapa kompartemen. Kompartemen adalah area dalam kelas yang berisi informasi. Kompartemen pertama berisi nama kelas, berikutnya atribut dan yang terakhir operasi, seperti pada gambar 1.



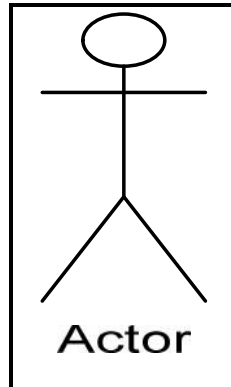
Gambar 1. Notasi Class

2. Diagram paket (*package diagram*)
3. Diagram use-case

Diagram ini digunakan untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. Use case menggambarkan *external view* dari sistem yang akan dibuat modelnya.

Komponen pembentuk diagram use case adalah:

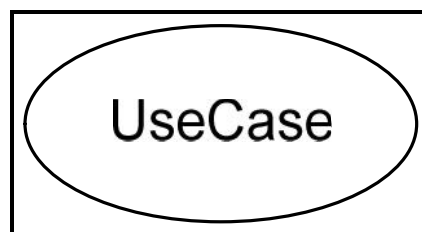
- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem, simbol aktor seperti pada gambar 2.



**Gambar 2.** Simbol Aktor

Ada empat macam tipe aktor:

- Primary business actor (aktor bisnis utama), yaitu stakeholder yang terutama mendapatkan keuntungan dari pelaksanaan use-case dengan menerima nilai yang terukur atau terobservasi.
  - Primary sistem actor (aktor sistem utama), yaitu stakeholder yang secara langsung berhadapan dengan sistem untuk menginisiasi atau memicu kegiatan atau sistem.
  - External server actor (aktor server eksternal) yaitu stakeholder yang melayani kebutuhan pengguna use-case.
  - External receiving actor (aktor penerima eksternal) yaitu stakeholder yang bukan pelaku utama, tapi menerima nilai yang terukur atau teramati.
- b. Use case, aktifitas/ sarana yang disiapkan oleh bisnis/ sistem. Use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Use case juga dapat diartikan sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. Use case digambarkan dalam bentuk elips atau oval, seperti pada gambar 3.



**Gambar 3.** Simbol Use Case.

- c. Hubungan (*link*), aktor mana saja yang terlibat dalam use case ini.
4. Diagram interaksi dan sequence
  5. Diagram komunikasi (communication diagram)
  6. Diagram statechart (statechart diagram)
- Diagram statechart dalam UML kadang disebut dengan istilah diagram state machine. Diagram ini menggambarkan perilaku sistem perangkat lunak yang kita buat dan perilaku

kelas, subsistem dan seluruh aplikasi. Selain itu diagram state machine bermanfaat juga untuk menyediakan cara yang baik dalam memodelkan komunikasi yang terjadi dengan entitas luar via protokol atau sistem dasarnya. UML memiliki dua state machine antara lain:

- Behavioral State Machine, mesin ini digunakan untuk menunjukkan perilaku elemen yang dimodelkan, misalnya suatu objek.
- Protocol State Machine, mesin ini digunakan untuk menunjukkan perilaku protokol.

Behavioral state machine dan protocol state machine saling melengkapi dan berbagi fungsi dengan batasan tertentu.

7. Diagram aktivitas (activity diagram)

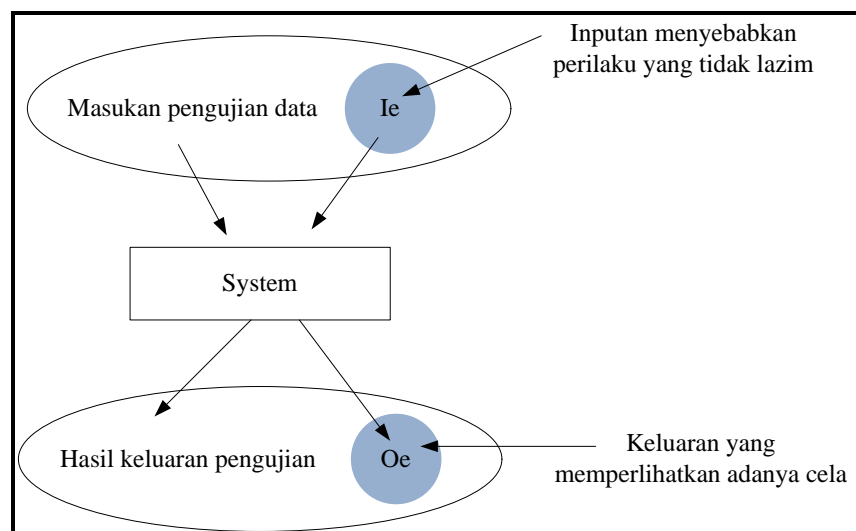
Diagram aktifitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

8. Diagram komponen (component diagram)

9. Diagram deployment (deployment diagram)

### Black Box Testing

Menurut Sommerville (2004), proses *black-box testing* adalah pengujian yang berasal dari spesifikasi sistem. Sistem diperlakukan seperti sebuah kotak hitam (*black-box*) yang mana perilaku hanya bisa ditentukan dengan mempelajari inputan dan keluaran yang berhubungan. Nama lainnya adalah pengujian fungsional karena penguji hanya dikonsentrasikan terhadap fungsionalitas dan tidak pengimplementasi dari perangkat lunak. Gambar 4 mengilustrasikan model dari sebuah sistem yang diasumsikan dalam pengujian black-box. Penguji memberikan masukan-masukan untuk komponen atau sistem dan memeriksa keluaran yang cocok. Jika keluaran tidak terprediksi, pengujian mendeteksi sebuah masalah dengan perangkat lunak tersebut.



Gambar 4. Pengujian Black-box

### **MVC (Model View Controller)**

Menurut Ardhana (2013), MVC merupakan sebuah pattern atau teknik pemrograman yang memisahkan antar pengembang aplikasi berdasarkan komponen utama pada sebuah aplikasi, seperti manipulasi data, user interface dan bagian yang menjadi kontrol aplikasi.

Secara sederhana dapat dikatakan bahwa antara desain dan proses data berada pada tempat yang terpisah. Saat ini MVC merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, yang berawal pada bahasa small talk. Terdapat tiga jenis komponen yang membangun suatu MVC pattern, yaitu:

#### **1. Model**

Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga mempresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, XML maupun webservice.

Biasanya didalam model akan berisi class dan fungsi untuk melakukan manipulasi data seperti insert, update, delete dan search, namun tidak dapat berhubungan dengan bagian view secara langsung, aplikasi website biasanya menggunakan database untuk menyimpan data, oleh karena itu model biasanya akan berhubungan dengan perintah-perintah query SQL.

#### **2. View**

View berhubungan dengan segala sesuatu yang akan ditempatkan ke end-user, biasa berupa halaman web, RSS, Javascript dan lain-lain. Programmer harus menghindari adanya logika pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan.

View dapat dikatakan sebagai halaman website yang dibuat menggunakan HTML dengan bantuan CSS, Javascript dan JQuery. Didalam view juga harus dihindari adanya kode untuk melakukan koneksi ke database. View hanya مخصوص untuk menampilkan data-data hasil dari model dan controller. Bagian ini tidak memiliki akses secara langsung terhadap bagian model.

#### **3. Controller**

Controller merupakan penghubung antara model dan view. Didalam controller inilah terdapat class dan fungsi-fungsi yang memproses permintaan dari view kedalam struktur data didalam model. Controller juga tidak boleh berisi kode untuk mengakses basis data.

Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke database, menyediakan validasi atau pengecekan terhadap input.

### **Pemrograman Beorientasi Objek**

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya (Shalahuddin dkk, 2011:82). Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

#### **1. Meningkatkan produktivitas**

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (reuseable).



2. Kecepatan pengembangan  
Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengolahan.
3. Kemudahan pemeliharaan  
Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Konsistensi  
Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Kualitas perangkat lunak  
Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.  
Beberapa contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek: bahasa pemrograman Smalltalk, Eiffel, C++, PHP, Java.

### **Analisis dan Desain Berorientasi Objek**

Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan untuk menganalisis spesifikasi atau kebutuhan akan sistem yang akan dibangun dengan konsep berorientasi objek. OOA biasanya menggunakan kartu CRC (Component, Responsibility, Collaborator) untuk membangun kelas-kelas yang akan digunakan atau menggunakan UML (Unified Modeling Language) pada bagian diagram use case, diagram kelas, dan diagram objek (Shalahuddin dkk, 2011:96).

Desain berorientasi objek atau *Object Oriented Design* (OOD) adalah tahapan perantara untuk memetakan spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan agar lebih mudah diimplementasikan dengan pemrograman berorientasi objek.

Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat pemodelan berorientasi objek, diantaranya adalah UML. Kendala dan permasalahan pembangunan sistem berorientasi objek biasanya dapat dikenali dalam tahap ini.

OOA dan OOD dalam proses yang berulang-ulang seringkali memiliki batasan yang samar, sehingga kedua tahapan ini sering juga disebut OOAD (Object Oriented Analysis and Design) atau dalam bahasa Indonesia berarti Analisis dan desain berorientasi objek.

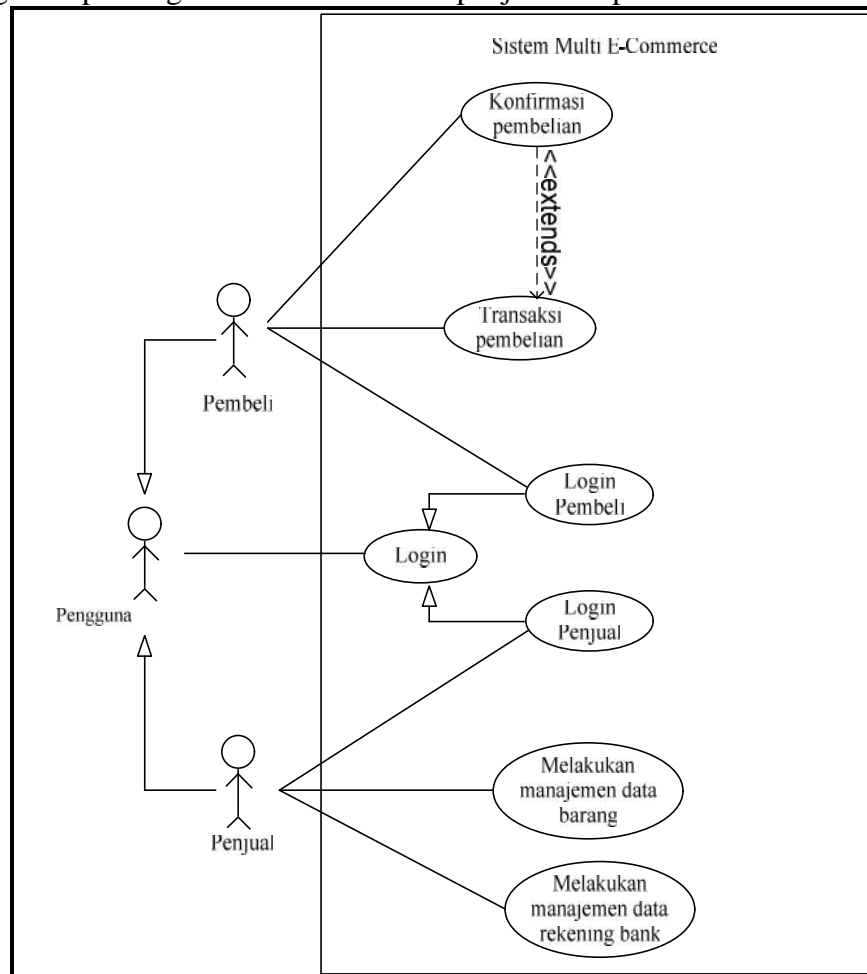
### **METODE PENELITIAN**

Dalam penelitian ini menggunakan teknik pengembangan prototyping, diterapkan dalam kerangka kerja penelitian tahap demi tahap dan menghasilkan sebuah website e-commerce berbentuk prototype berbasis OOP, dibangun menggunakan arsitektur MVC dan diterjemahkan menggunakan framework Codeigniter. Pada tahap analisis dan desain pada fase prototyping menggunakan alat analisis-desain Unified Modelling Language (UML), dan aplikasi diuji dengan

menggunakan teknik black box untuk memastikan secara fungsional prototype dapat berjalan dengan baik.

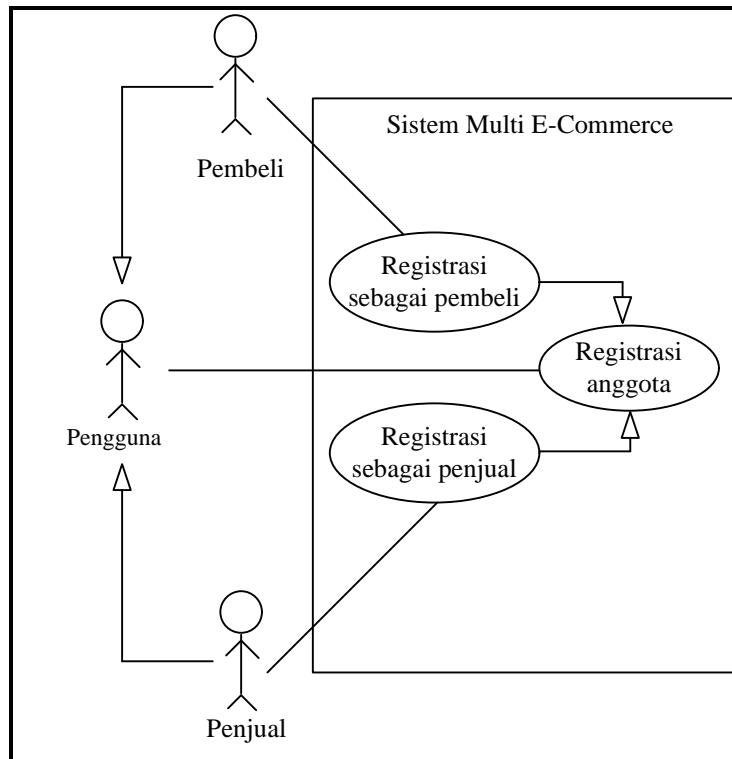
## HASIL DAN PEMBAHASAN

Pada gambar 5, 6 dan 7 adalah diagram use case untuk memodelkan perilaku sistem. Perilaku sistem yang didapat dari pengamatan langsung beberapa sistem e-commerce yang ada, dan dijelaskan dalam bentuk diagram use case. Pelaku atau aktor yang terlibat dalam sistem adalah pengguna yang merupakan generalisasi dari aktor penjual dan pembeli.



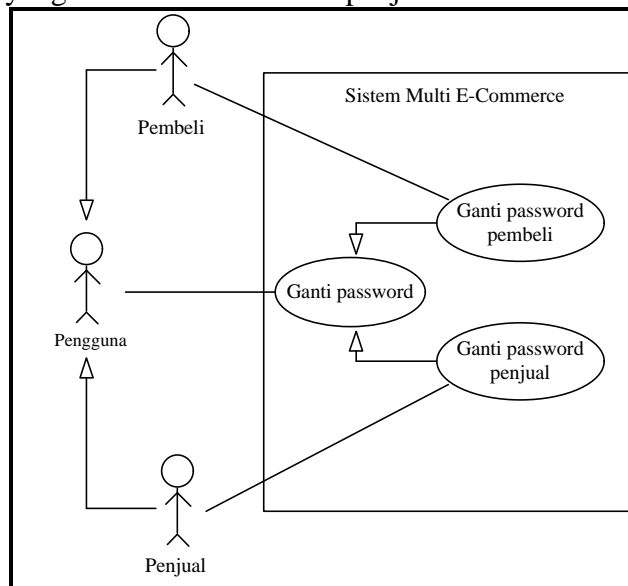
**Gambar 5.** Diagram Use Case Pemodelan Perilaku Sistem

Pada gambar 6 adalah diagram use case yang menjelaskan tentang proses registrasi yang dimiliki oleh sipengguna. Registrasi adalah use case yang bersifat general, karena bisa dibedakan untuk registrasi untuk penjual maupun registrasi untuk pembeli.



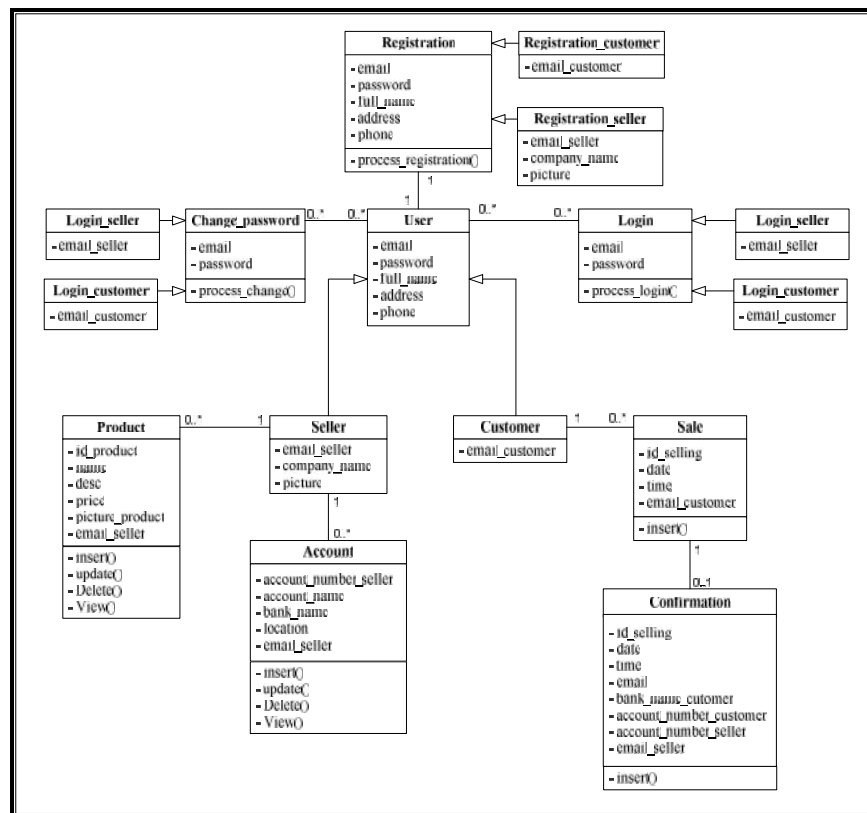
**Gambar 6.** Diagram Use Case Pemodelan Perilaku Sistem

Pada gambar 7 menjelaskan kondisi generalisasi untuk use case ganti password. Use case ganti password dapat diturunkan menjadi ganti password yang dilakukan oleh aktor pembeli maupun ganti password yang dilakukan oleh aktor penjual.



**Gambar 7.** Diagram Use Case Pemodelan Perilaku Sistem

Berdasarkan diagram use case pada gambar 5, 6 dan 7, dapat diturunkan menjadi sebuah diagram class. Diagram class pada gambar 4 menjelaskan hubungan class satu sama lain, dengan derajat hubungan, atribut yang dimiliki setiap class yang dapat dikonversi menjadi susunan table database yang digunakan untuk menyimpan data, dan metode class yang dapat digunakan untuk menjelaskan fungsional aplikasi yang akan dibangun.



**Gambar 8.** Diagram Class Sistem

Berdasarkan diagram class pada gambar 8 maka dapat diturunkan ke dalam desain tabel beserta field penyusunnya.

**Table 1.** seller dengan primary key: email\_seller, dan foreign key: -

No	Nama Field	Tipe data	Panjang	Keterangan
1	email_seller	Text	60	Not Null
2	company_name	Text	100	Not Null
3	Picture	Text	100	Null
4	full_name	Text	60	Not Null
5	Password	Text	60	Not Null

6	Address	Text	100	Not Null
7	Phone	Text	100	Null

**Table 2. : costumer dengan primary key: email\_costumer, dan foreign key: -**

No	Nama Field	Tipe data	Panjang	Keterangan
1	email_costumer	Text	60	Not Null
2	Password	Text	60	Not Null
3	full_name	Text	100	Not Null
4	address	Text	100	Not Null
5	Phone	Text	60	Null

**Table 3. : Product dengan primary key: id\_product, dan foreign key: email\_seller**

No	Nama Field	Tipe data	Panjang	Keterangan
1	id_product	Text	10	Not Null
2	Name	Text	100	Not Null
3	Desc	Text	100	Not Null
4	Price	Number	10	Not Null
5	picture	Text	100	Null
6	email_seller	Text	60	Not Null

**Table 4. : sale dengan primary key: id\_selling, dan foreign key: email\_costumer**

No	Nama Field	Tipe data	Panjang	Keterangan
1	id_selling	Text	20	Not Null
2	Date	Text	10	Not Null
3	Time	Text	10	Not Null
4	email_customer	Text	60	Not Null

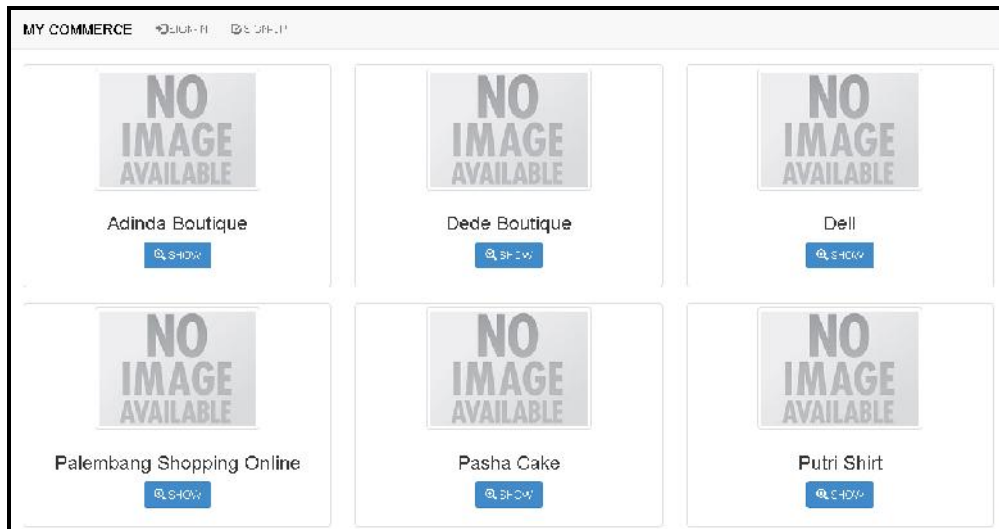
**Table 5. : account dengan primary key: account\_number\_seller, dan foreign key: email\_seller**

No	Nama Field	Tipe data	Panjang	Keterangan
1	account_number_seller	Text	20	Not Null
2	account_name	Text	100	Not Null
3	bank_name	Text	100	Not Null
4	location	Text	100	Nut Null
5	email_seller	Text	60	Not Null

**Table 6.: confirmation dengan primary key: id\_selling, dan foreign key: email\_seller, account\_number\_seller**

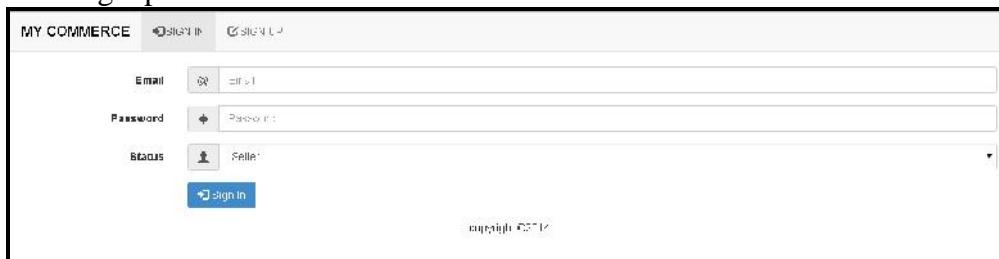
No	Nama Field	Tipe data	Panjang	Keterangan
1	id_selling	Text	10	Not Null
2	Date	Text	10	Not Null
3	Time	Text	10	Not Null
4	bank_name_customer	Text	100	Not Null
5	account_number_customer	Text	20	Not Null
6	account_number_seller	Text	20	Not Null
7	email_seller	Text	60	Not Null

Pada gambar 9 adalah tampilan halaman utama yang menampilkan daftar nama penjual. Halaman ini adalah halaman yang pertama akan ditampilkan ketika kita mengakses website.



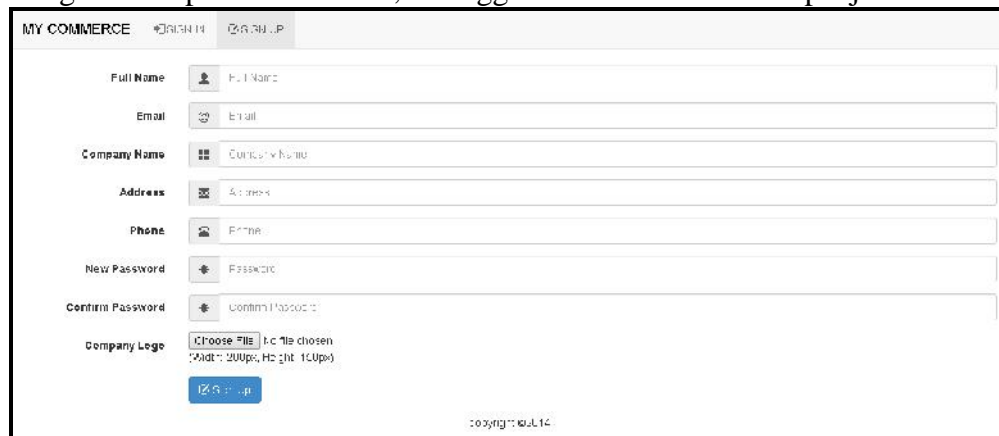
**Gambar 9.** Tampilan Halaman Utama

Pada gambar10 adalah tampilan halaman login, halaman login digunakan oleh pengguna untuk masuk ke dalam menu utama. Pada halaman login terdapat pilihan untuk masuk sebagai penjual atau sebagai pembeli.



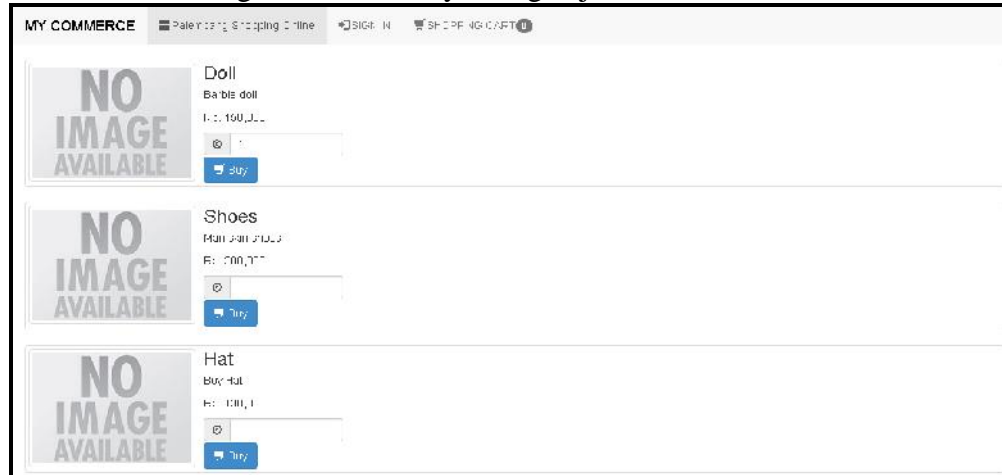
**Gambar 10.** Tampilan Halaman Login

Pada gambar 11 adalah tampilan halaman registrasi untuk para pengguna yang akan menjual barang mereka pada website ini, sehingga status mereka adalah penjual.



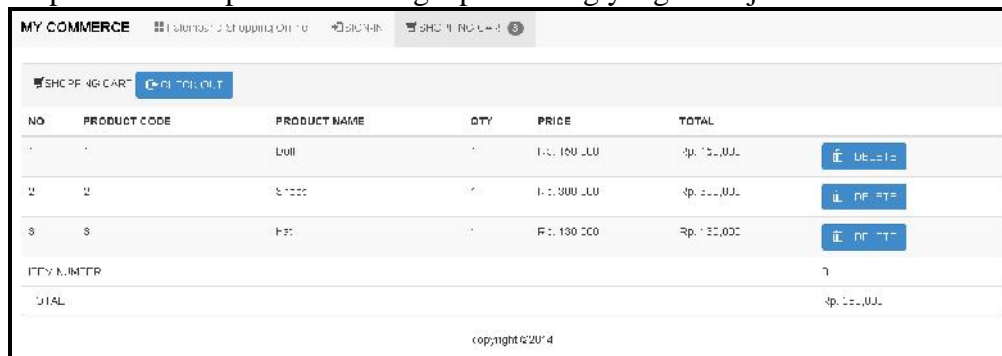
**Gambar 11.** Tampilan Halaman Register Penjual

Pada gambar 12 adalah tampilan halaman transaksi penjualan dan pembelian, dimana para pembeli bisa memilih barang dan membelinya dengan jumlah tertentu.



Gambar 12. Tampilan Halaman Transaksi

Pada gambar 13 adalah tampilan halaman keranjang belanja, dimana barang-barang yang dipilih akan dikumpulkan dalam keranjang belanja sebelum disahkan untuk dibeli. Pada halaman ini juga terdapat tombol hapus untuk menghapus barang yang tidak jadi untuk dibeli.



Gambar 13. Tampilan Halaman Shopping Cart

Pada gambar 14 adalah tampilan halaman untuk mengisi data pembeli, ketika menekan tombol checkout pada halaman keranjang belanja.





**Gambar 14.** Tampilan Halaman Register Pembeli

Pada gambar 15 adalah tampilan halaman menu utama untuk pembeli.



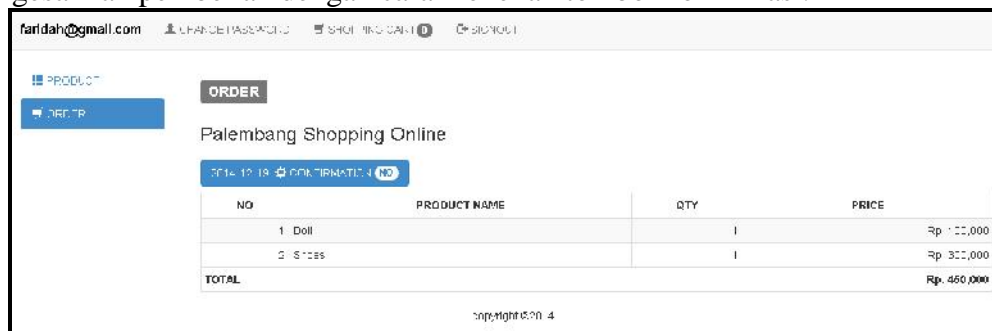
**Gambar 15.** Tampilan Halaman Menu Utama Pembeli

Pada gambar 16 adalah tampilan halaman ganti password. Halaman ganti password terdapat untuk pengguna dengan status pembeli dan penjual.



**Gambar 16.** Tampilan Halaman Ganti Password

Pada gambar 17 adalah tampilan halaman daftar barang yang telah dipesan oleh pembeli, untuk mengesahkan pembelian dengan cara menekan tombol konfirmasi.



NO	PRODUCT NAME	QTY	PRICE
1	Doll	1	Rp. 100,000
2	Stress	1	Rp. 300,000
TOTAL			Rp. 400,000

**Gambar 17.** Tampilan Halaman Data Pesanan Barang

Pada gambar 18 adalah tampilan halaman konfirmasi pembelian, pembeli mengisi data transfer uang yang telah dilakukan si pembeli.

The screenshot shows a web application interface for purchase confirmation. On the left, there is a sidebar with a 'PRODUCT' link and an 'ORDER' link. The main content area is titled 'CONFIRMATION'. It contains several form fields: 'Your BANK NAME' (with a placeholder 'Contoh: BANK ABC'), 'Your Account Number' (with a placeholder 'Contoh: 0123456789'), 'Your Account Name' (with a placeholder 'Contoh: Yudi Filantini'), and 'To Account' (with a dropdown menu showing 'EAT - SH - T - Octafian Palembang'). A 'Send' button is located at the bottom of the form. The footer of the page indicates 'Copyright © 2014'.

**Gambar 18.** Tampilan Halaman Konfirmasi Pembelian

Pada gambar 19 adalah tampilan halaman menu utama untuk penjual.

The screenshot shows the main menu page for a seller. On the left, there is a sidebar with three links: 'PRODUCT', 'CONFIRMATION', and 'ACCOUNT'. The main content area is mostly blank, with a 'Copyright © 2014' footer.

**Gambar 19.** Tampilan Halaman Menu Utama Penjual

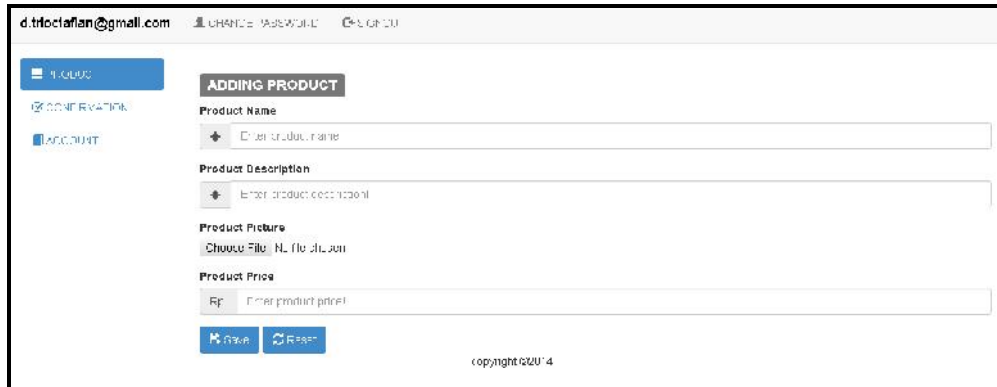
Pada gambar 20 adalah tampilan halaman daftar barang yang akan dijual.

The screenshot shows the product list page for a seller. On the left, there is a sidebar with three links: 'PRODUCT', 'CONFIRMATION', and 'ACCOUNT'. The main content area displays a table of products. The table has columns for 'ID', 'PRODUCT', 'DESCRIPTION', 'PRICE', and action buttons ('EDIT' and 'DELETE').

ID	PRODUCT	DESCRIPTION	PRICE	
1	Cell	Cell phone from animal skin	Rp. 120,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
2	Drill	Drill	Rp. 150,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
3	Glove	Shin Glove	Rp. 70,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
4	Handkerchief	Handkerchief from animal skin	Rp. 100,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
5	Hat	Easy Hat	Rp. 120,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
6	Basket	Ancient basket from animal skin	Rp. 100,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
7	Ring	Ring for women from animal skin	Rp. 100,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
8	Socks	Socks for women from animal skin	Rp. 100,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
9	Shawl	Shawl from animal skin	Rp. 100,000	<a href="#">EDIT</a> <a href="#">DELETE</a>
10	Shawl	Shawl from animal skin	Rp. 100,000	<a href="#">EDIT</a> <a href="#">DELETE</a>

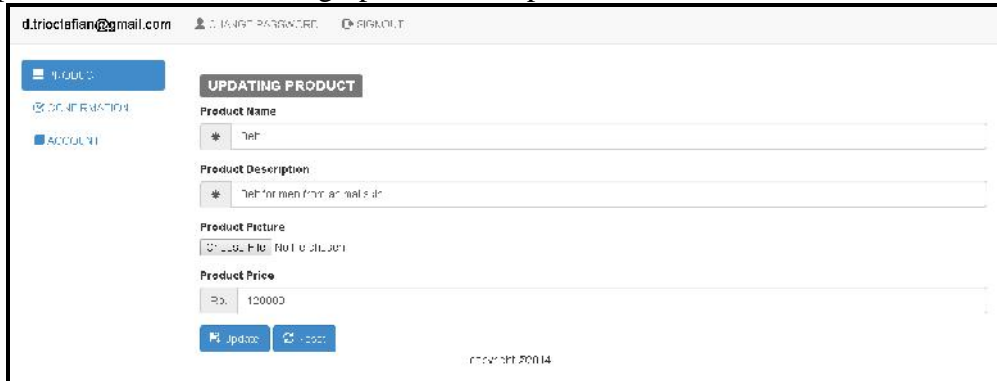
**Gambar 20.** Tampilan Halaman Data Barang Pada Penjual

Pada gambar 21 adalah tampilan halaman untuk menambah data barang yang akan dijual.




**Gambar 21.** Tampilan Halaman Tambah Data Barang Pada Penjual

Pada gambar 22 adalah tampilan halaman untuk mengubah data barang yang akan dijual, jika terdapat kesalahan ketika menginput data saat proses tambah data.



**Gambar 22.** Tampilan Halaman Ubah Data Barang Pada Penjual

Pada gambar 23 adalah tampilan halaman untuk menghapus data barang jika informasi tidak diperlukan lagi oleh penjual. Terdapat tombol yes dan no untuk melakukan konfirmasi penghapusan data.



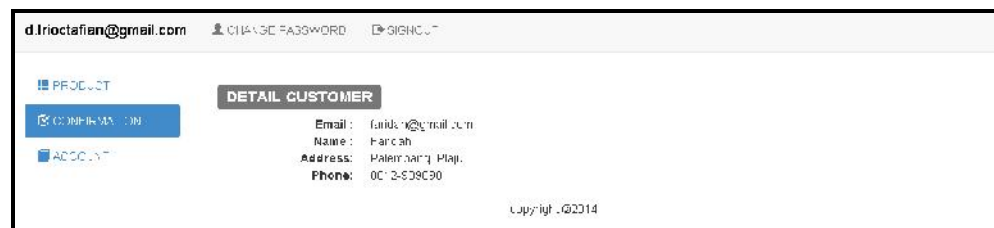
**Gambar 23.** Tampilan Halaman Hapus Data Barang Pada Penjual

Pada gambar 24 adalah tampilan halaman daftar barang yang telah dipesan dan dikonfirmasi oleh pembeli, sehingga barang siap untuk diantarkan ke pembeli.



**Gambar 24.** Tampilan Halaman Data Konfirmasi Pembelian Pada Penjual

Pada gambar 25 adalah tampilan halaman informasi sipembeli yang ada pada fasilitas sipenjual.



**Gambar 25.** Tampilan Halaman Informasi Pembeli

Pada gambar 26 adalah tampilan halaman detail barang yang telah dikonfirmasi pemesanan oleh sipembeli.



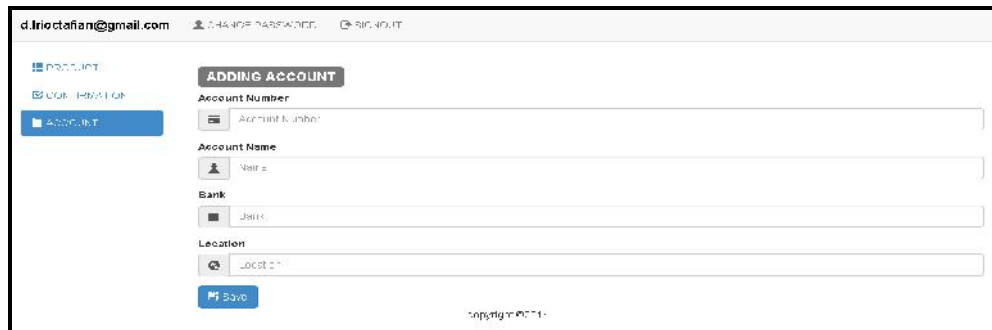
**Gambar 26.** Tampilan Halaman Detail Pesanan Pada Penjual

Pada gambar 27 adalah tampilan halaman daftar rekening yang dimiliki oleh sipenjual.



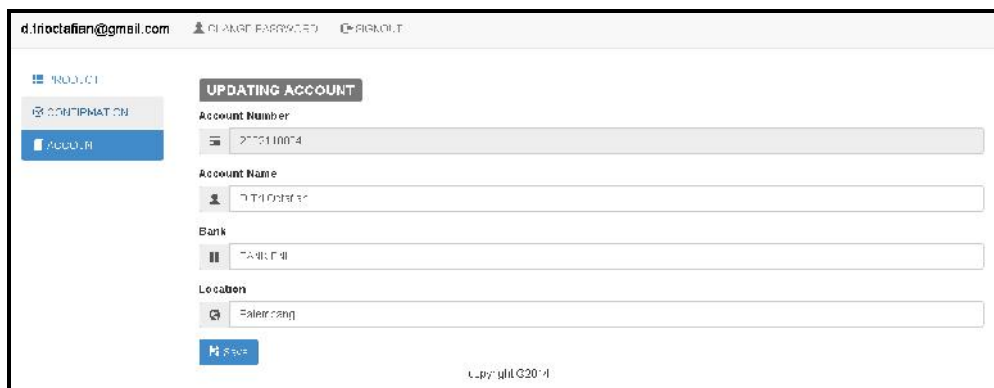
**Gambar 27.** Tampilan Halaman Data Rekening Bank Penjual

Pada gambar 28 adalah tampilan halaman untuk menambah data rekening bank yang dimiliki oleh penjual.

The screenshot shows a web application interface for adding a new bank account. On the left, there is a sidebar with a menu containing 'PRODUCT', 'CONFIRMATION', and 'ACCOUNT'. The 'ACCOUNT' menu item is highlighted with a blue background. The main content area is titled 'ADDING ACCOUNT' in a dark header. Below the header, there are four input fields: 'Account Number' (with a card icon), 'Account Name' (with a person icon), 'Bank' (with a building icon), and 'Location' (with a location pin icon). Each field has a corresponding icon on the left. At the bottom of the form, there is a blue 'Save' button. The footer of the page contains the text 'Copyright © 2014'.

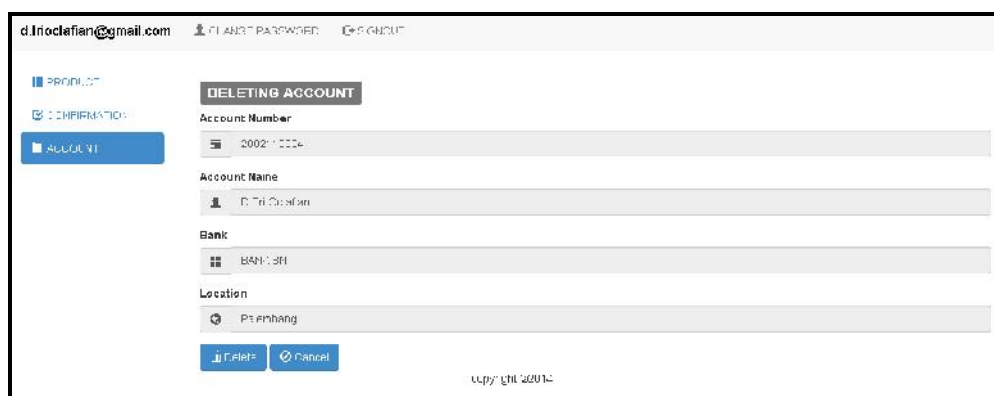
**Gambar 28.** Tampilan Halaman Tambah Data Rekening Bank Penjual

Pada gambar 29 adalah tampilan halaman untuk mengubah data rekening bank yang telah diinputkan oleh penjual, jika mengalami kesalahan ketika data ditambahkan.

The screenshot shows a web application interface for updating an existing bank account. The sidebar menu is the same as in the previous image, with 'ACCOUNT' highlighted. The main content area is titled 'UPDATING ACCOUNT'. The form fields are identical to the 'ADDING ACCOUNT' form, but they are pre-filled with data: 'Account Number' is '2772110014', 'Account Name' is 'D. Tri Octafian', 'Bank' is 'Mandiri', and 'Location' is 'Palembang'. A blue 'Save' button is at the bottom. The footer text is 'Copyright © 2014'.

**Gambar 29.** Tampilan Halaman Ubah Data Rekening Bank Penjual

Pada gambar 29 adalah tampilan halaman untuk menghapus data rekening yang tidak terpakai lagi oleh sipenjual.

The screenshot shows a web application interface for deleting a bank account. The sidebar menu is the same, with 'ACCOUNT' highlighted. The main content area is titled 'DELETING ACCOUNT'. The form fields are pre-filled with data: 'Account Number' is '2002110014', 'Account Name' is 'D. Tri Octafian', 'Bank' is 'BANK BRI', and 'Location' is 'Pekalongan'. At the bottom, there are two buttons: a blue 'Delete' button and a grey 'Cancel' button. The footer text is 'Copyright © 2014'.

**Gambar 29.** Tampilan Halaman Hapus Data Rekening Bank Penjual

## **PENUTUP**

Berdasarkan penelitian yang telah dilakukan, maka beberapa kesimpulan yang dapat diambil, yaitu: pengguna dengan status penjual dapat memiliki website e-commerce dengan melakukan registrasi terlebih dahulu, kemudian dapat menambahkan data produk yang akan dijual dan pengguna dengan status pembeli dapat melakukan transaksi pada website e-commerce, dengan melakukan pemesanan terlebih dahulu, mengisi data pembeli, mentransfer uang sesuai dengan jumlah yang dipesan, dan melakukan konfirmasi pembayaran.

## **DAFTAR PUSTAKA**

- Triton PB. 2006. *Mengenal E-commerce dan Bisnis Di Dunia Cyber*. Argo: Yogyakarta.
- Pressman, Roger. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Andi : Yogyakarta.
- Shalahuddin, M & A.S, Rosa. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Modula: Bandung.
- Ardhana, YM Kusuma. 2013. *Pemrograman PHP: Codeigniter Black Box*. Jasakom :Jakarta.
- Basuki, Awan. 2010. *Membangun Web Berbasis PHP dengan Framework Codeigniter*. Lokommedia: Yogyakarta.
- Widodo, Prabowo Pudjo & Herlawati. 2011. *Menggunakan UML*. Informatika, Bandung.
- Sommerville, Ian. 2004. *Software Engineering: Seventh Edition*. Pearson Education Limited: England.