

# IMPLEMENTASI *LIBRARY ACTIVE MERCHANT* SEBAGAI *VIRTUAL BANK* DALAM APLIKASI *E-COMMERCE*

Achmad Zakki Falani, S.Kom, M.Kom.<sup>1</sup>, Muhammad Hidayatullah<sup>2</sup>,

<sup>1</sup>Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Narotama Surabaya,

<sup>2</sup>Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Narotama Surabaya

## ABSTRAK

Perkembangan internet menyebabkan terbentuknya sebuah dunia baru yang lazim disebut dunia maya. Di dunia maya ini setiap individu memiliki hak dan kemampuan untuk berinteraksi dengan individu lain tanpa batasan apapun yang dapat menghalanginya. Seluruh aspek kehidupan manusia terkena dampak kehadiran internet, sektor bisnis merupakan sektor yang paling terkena dampak dari perkembangan teknologi informasi dan telekomunikasi serta paling cepat tumbuh. Dalam dunia *e-commerce* pelaku bisnis diwajibkan memiliki rekening di dalam *virtual bank* untuk memudahkan dan mempercepat transaksi pembayaran. *Ruby on rails* merupakan bahasa pemrograman dengan memiliki berbagai macam fitur didalam pengembangan aplikasi, khususnya *library active merchant* yang dapat menjawab tantangan serta kebutuhan dalam transaksi pembayaran di dalam dunia *e-commerce*.

**Kata kunci:** *e-commerce*, *library active merchant*, *ruby on rails*.

### 1.1 Latar Belakang

Penggunaan teknologi informasi pada era globalisasi sangat berkembang dengan pesat. Internet menjadi suatu teknologi yang paling berpengaruh dalam kehidupan sehari-hari. Menurut David Baum (1999, pp. 36-34) *E-Commerce* merupakan satu set dinamis teknologi, aplikasi, dan proses bisnis yang menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik dan perdagangan barang, pelayanan, dan informasi yang dilakukan secara elektronik. Dalam aplikasi web dapat terjadi transaksi secara *online*. Dengan adanya transaksi secara *online* di *e-commerce*, maka dibutuhkan *virtual bank* didalam aplikasi. Menurut Adjie (2011) *virtual bank* adalah Lembaga keuangan berbasis internet yang menawarkan fasilitas deposit dan penarikan, dan jasa perbankan lainnya, melalui anjungan tunai mandiri atau perangkat lain, tanpa harus memiliki perangkat fisik.

Pelaku bisnis *online* diwajibkan mempunyai rekening di *bank virtual* untuk

memudahkan transaksi. Di *bank virtual* tersebut pelaku bisnis dapat menyimpan uang yang berasal dari pendapatan bisnis *online* yang pelaku bisnis jalani atau deposit dari *bank* nyata yang telah terhubung (*linked*) dengan bank virtual tersebut. Pelaku bisnis juga dapat mengambil uang (*withdraw*) saldo di rekening di *bank virtual* ke bank yang telah terhubung dengan *bank virtual* tersebut.

### 1.2 Rumusan Masalah

Dalam penelitian ini nantinya terdapat beberapa rumusan masalah diantaranya sebagai berikut:

1. Bagaimana merancang toko *online* menggunakan *Unified Modeling Language (UML)* ?
2. Bagaimana pemanfaatan *library active merchant* dari *Shopify* sebagai *virtual bank* untuk pembayaran *online* ?
3. Bagaimana membangun toko *online* menggunakan bahasa pemrograman *ruby on rails* ?

4. Bagaimana membuat sistem dengan berbagai macam fitur yang tersedia di dalam *e-commerce*?

### 1.3 Batasan Masalah

Supaya sistem tidak melebar dan penelitian ini nantinya lebih terarah maka perlu adanya batasan masalah dalam mencakup sistem keseluruhan.

1. Kebutuhan Input yang meliputi:

- Data katalog produk yang berguna untuk mendata produk yang akan dijual di toko *online*. Atribut data katalog produk terdiri dari nama produk, kode produk, harga, *stock*, *size*, keterangan.
- Data *member* yang berguna untuk mendata *member* yang sudah daftar di toko *online*. Atribut data member terdiri dari nama depan, nama belakang, alamat, email, telepon, provinsi, kota.
- Data cakupan pengiriman berdasarkan masing-masing wilayah yang berguna untuk memberikan informasi biaya pengiriman, berat *item*, dan lama hari pengiriman.
- Data *review* produk yang berguna untuk *review* dari *user* yang sudah pernah membeli produk tersebut.

2. Proses yang meliputi:

- Manajemen data katalog produk yang berfungsi untuk menambah produk, mengubah produk, dan menghapus produk.
- Manajemen data *member* yang berfungsi untuk menambah data *member*, mengubah data *member*, dan menghapus data *member*.
- Manajemen data ongkos kirim berdasarkan tiap wilayah yang berfungsi untuk menambah data ongkos kirim, mengubah data ongkos kirim, dan menghapus data ongkos kirim.
- Transaksi pembayaran *online* berfungsi untuk pembayaran barang yang telah dipesan.
- Pemesanan produk berfungsi untuk

mengetahui *customer* yang telah memesan produk.

- Pendaftaran *member* berfungsi untuk bisa melakukan *checkout* dan pembayaran.
  - Pembuatan laporan pemesanan berfungsi untuk mencetak data laporan pemesanan barang.
3. Kebutuhan Output, meliputi:
    - Laporan pemesanan.
    - Produk yang terlaris (*top product*).
    - Produk yang termahal.
    - Produk yang termurah.
    - Produk yang banyak dilihat (*hit rank*).
  4. Perancangan toko online menggunakan alat bantu berupa :
    - *ArgoUML* versi 0.34 untuk membuat *Unified Modeling Language (UML)* versi 1.4 yang terdiri dari empat macam diagram, yaitu: Use Case Diagram, Class Diagram, Activity Diagram, Sequence Diagram.
    - *GUI Design* versi 4.2.115.0 untuk membuat desain *input / output*.
  5. Implementasi toko online menggunakan alat bantu berupa:
    - *MySQL* versi 5 yang berguna sebagai *database server* untuk menyimpan data-data yang telah dimasukkan.
    - *Ruby* versi 1.9.3 sebagai bahasa pemrograman dan *rails* versi 3.2 sebagai *framework ruby*.
    - *Javascript* untuk *script tag* yang berfungsi untuk tambahan bahasa pemrograman di aplikasi agar halaman web menjadi hidup.
    - *Sublime 2* yang berguna sebagai *text editor* untuk membuat *script* toko *online*.
    - *Library active merchant* dari Shopify sebagai *virtual bank* untuk pembayaran *online*.
    - Cakupan pengiriman *order* barang seluruh Indonesia.

- Mata uang yang digunakan untuk pembayaran adalah Rupiah (Rp) yang kemudian diubah menjadi mata uang Dollar Amerika (US\$). Hal ini dilakukan karena pembayaran dengan mata uang Rupiah (Rp) belum didukung oleh *sandbox paypal*.

#### 1.4 Tujuan dan Manfaat

Tujuan dari penelitian ini nantinya adalah:

1. Untuk merancang toko *online* menggunakan *Unified Modeling Language (UML)*.
2. Untuk memanfaatkan *library active merchant* dari Shopify sebagai *virtual bank* untuk pembayaran *online* dalam toko *online*.
3. Untuk membuat toko *online* menggunakan bahasa pemrograman *ruby on rails*.
4. Untuk membuat sistem toko *online* dengan berbagai macam fitur yang disediakan.

Manfaat dari penelitian ini adalah memudahkan pembayaran secara *online* dengan menggunakan fitur *sandbox paypal* yang sudah terdapat pada sistem serta memberikan pelayanan yang cepat dan tepat terhadap *customer* dikarenakan informasi yang *up to date*.

#### 2.1 Ruby on Rails

*Ruby* adalah bahasa pemrograman *open source* dinamis, mudah dimengerti dan produktif. Sintaks *ruby* elegan, natural, mudah dibaca dan ditulis. Pencipta *ruby*,

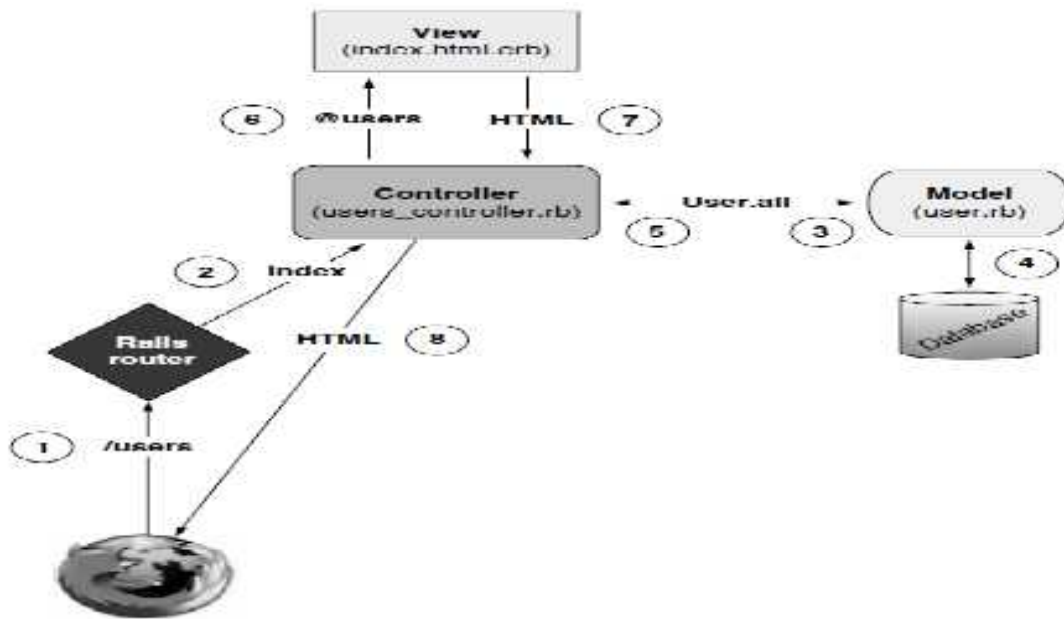
Yukihiro “matz” Matsumoto, menggabungkan bagian-bagian dari bahasa-bahasa favorit beliau (*Perl*, *Smalltalk*, *Eiffel*, *Ada* dan *Lisp*) untuk membentuk bahasa baru yang seimbang antara pemrograman fungsional dengan pemrograman imperatif. Inti dari bahasa pemrograman *ruby* yaitu semua objek.

*Rails* merupakan *ruby-based framework* yang menggunakan pendekatan *Model-View-Controller (MVC)*.

*Rails* merupakan salah satu *framework ruby* yang paling populer sekarang ini. Oleh karena itu dinamakan *Ruby on Rails (RoR)*. Dilihat dari namanya sering ada salah pengertian bahwa bukan *ruby* yang digunakan pada *rails* tetapi *rails* yang digunakan pada *ruby*.

*Rails* menggunakan *Model-View-Controller (MVC)* sebagai *design pattern*-nya. Struktur penyusun *MVC* ada 3 jenis yang saling bekerjasama, yaitu:

1. *Model* : ini berhubungan dengan *database*, yaitu *Structured Query Language (SQL)* dan untuk *validation*.
2. *View* : ini berhubungan dengan *User Interface* atau bisa dibayangkan tampilan suatu aplikasi tersebut di browser.
3. *Controller* : fungsi dari *controller* sendiri yaitu mengatur dari aplikasi. Keputusan mengenai *view* mana yang harus dipanggil untuk menampilkan data tertentu, bagian mana dari *model* yang harus *update* ditangani oleh *controller*. Berperan sebagai perantara dari aplikasi dan mengambil tiap *request*.



Gambar 1. Detail diagram Model-View-Controller rails

## 2.2 Object Oriented Analysis

*Object Oriented Analysis* (OOA) adalah mencari kebutuhan dari perspektif *class* dan *object* yang ditemukan dalam suatu *vocabulary* dari domain masalah. Dengan kata lain, *system* dimodelkan dalam bentuk *object* dan *class*.

## 2.3 Object Oriented Design

*Object Oriented Design* (OOD) adalah dekomposisi *Object Oriented* (OO) dan suatu notasi untuk menggambarkan model sistem pada tahap pengembangan. Struktur dibentuk setelah *object* yang berhubungan dengan sistem sudah didefinisikan.

## 2.4 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak.

*Unified Modeling Language* (UML) mempunyai beberapa diagram, yaitu:

1. *Use Case Diagram* dibuat untuk memvisualisasikan / menggambarkan hubungan antara *actor* dan *use case*. *Use case diagram* mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.
2. *Class Diagram* adalah *diagram* yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. *Class Diagram* memiliki beberapa simbol.
3. *Activity diagram* adalah menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* mempunyai beberapa simbol.
4. *Sequence diagram* adalah penjelasan proses yang terjadi pada sistem dalam menerima masukan (*input*) dari pengguna sampai dengan tersajinya informasi yang diminta oleh pengguna. *Sequence diagram* mempunyai beberapa simbol.

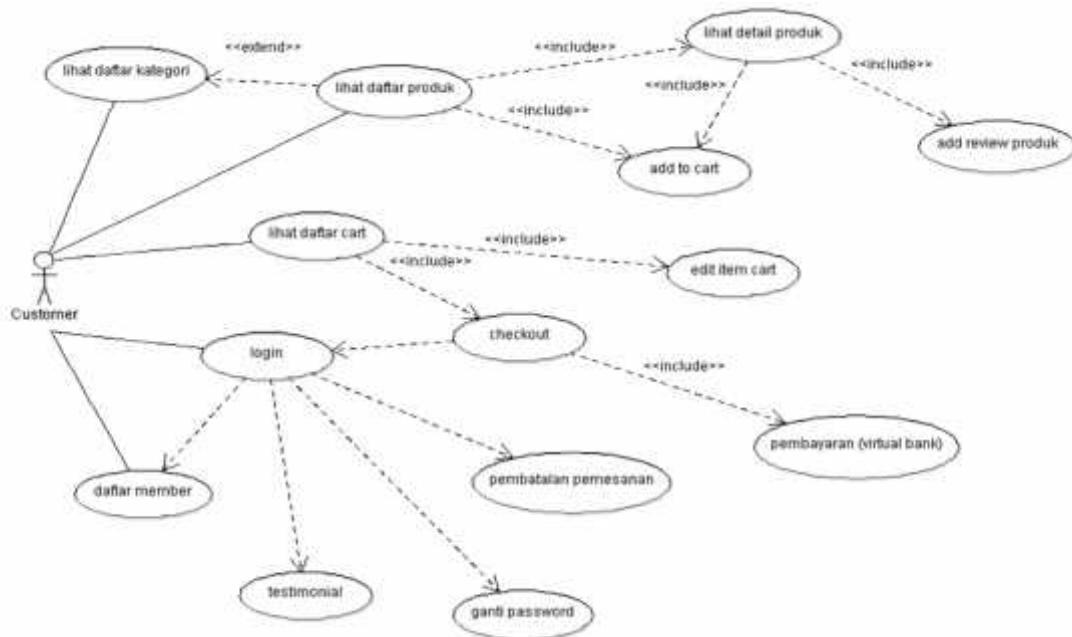
## 2.5 Library Active Merchant

*Library active merchant* adalah ekstraksi solusi *software* e-commerce dari Shopify. *Library active merchant* dikelola oleh perusahaan yang bernama Shopify dan

Spredly yang dikembangkan untuk aplikasi web menggunakan *ruby on rails* yang berhubungan dengan transaksi keuangan.

## 3.1 Perancangan Sistem

### 3.1.1 Use Case Diagram Customer



Gambar 2. Use Case Diagram Customer

Tabel 1. Naratif Use Case Lihat Daftar Kategori

<i>Use Case name</i> : <b>Lihat Daftar Kategori</b>
<i>Primary actor</i> : Customer
<i>Brief description</i> : Use case ini digunakan untuk melihat daftar kategori yang ada.
<i>Trigger</i> : -
<i>Type</i> : Eksternal
<i>Relationships</i> :
<i>Include</i> : -
<i>Exclude</i> : lihat daftar produk
<i>Normal flow of events</i> :
<ol style="list-style-type: none"> <li>1. Aktor mengunjungi aplikasi toko <i>online</i>.</li> <li>2. Sistem akan menampilkan halaman "Home".</li> <li>3. Aktor memilih tab "Kategori".</li> <li>4. Sistem menampilkan daftar kategori.</li> </ol>

Tabel 2. Naratif Use Case Lihat Daftar Kategori

<i>Use Case name</i> : <b>Lihat Daftar Keranjang Belanja</b>
<i>Primary actor</i> : Customer
<i>Brief description</i> : Use case ini digunakan untuk melihat daftar keranjang belanja yang ada.
<i>Trigger</i> : -
<i>Type</i> : Eksternal
<i>Relationships</i> :
<i>Include</i> : edit item cart, checkout, pembayaran
<i>Normal flow of events</i> : <ul style="list-style-type: none"> <li>1. Aktor mengunjungi aplikasi toko <i>online</i>.</li> <li>2. Sistem akan menampilkan halaman “Home”.</li> <li>3. Sistem menampilkan daftar produk.</li> </ul>

Tabel 3. Naratif *use case* lihat daftar keranjang belanja

<i>Use Case name</i> : <b>Lihat Daftar Produk</b>
<i>Primary actor</i> : Customer
<i>Brief description</i> : Use case ini digunakan untuk melihat daftar produk yang ada.
<i>Trigger</i> : -
<i>Type</i> : Eksternal
<i>Relationships</i> :
<i>Include</i> : edit item cart, checkout, pembayaran
<i>Normal flow of events</i> : <ul style="list-style-type: none"> <li>1. Aktor mengunjungi aplikasi toko <i>online</i>.</li> <li>2. Sistem akan menampilkan halaman “Home”.</li> <li>3. Aktor memilih tab “Keranjang belanja”.</li> <li>4. Sistem menampilkan daftar item yang ada di keranjang belanja.</li> <li>5. Customer bisa mengubah quantity item.</li> <li>6. Customer bisa melakukan checkout kemudian melakukan pembayaran.</li> </ul>
<i>Alternate/exceptional flows</i> :
6a Customer bisa melakukan checkout dan pembayaran jika sudah login di aplikasi.

Tabel 3. Naratif *use case* login

<i>Use Case name</i> : <b>Login</b>
<i>Primary actor</i> : Customer
<i>Brief description</i> : Use case ini digunakan customer untuk login di aplikasi.
<i>Trigger</i> : -
<i>Type</i> : Eksternal
<i>Relationships</i> :
<i>Include</i> : -

<i>Dependency</i> : daftar <i>member</i> , testimonial, ganti password, pembatalan pemesanan
<i>Normal flow of events</i> :
<ol style="list-style-type: none"> <li>1. Aktor mengunjungi aplikasi toko <i>online</i>.</li> <li>2. Sistem akan menampilkan halaman "Home".</li> <li>3. Aktor memilih list login di tab "<i>My Account</i>".</li> <li>4. Sistem menampilkan halaman login.</li> </ol>
<i>Alternate/exceptional flows</i> :

### 3.1.2 Use Case Diagram Admin



Gambar 3. Use Case Diagram Admin

#### 4.1 Requirement Hardware

Spesifikasi hardware yang dibutuhkan untuk *development* aplikasi dan implementasi *library active merchant* adalah

1. *Operating System* menggunakan *Linux*.
2. *Memory* minimum 1GB. *Processor* minimum *dual core*.

#### 4.2 Requirement Software

Spesifikasi software dan *library* yang dibutuhkan untuk *development* aplikasi adalah

1. *Install Ruby on Rails*.
2. *Library mysql2* yang berfungsi untuk mengkonekan ke database *mysql*.
3. *Library activemerchant* yang berfungsi untuk pembayaran online.
4. *Library friendly\_id* yang berfungsi untuk merubah id menjadi nama.

## Daftar Pustaka

Chang Sau Sheong. 2010. "*Ruby on Rails Web Mashup Projects*". Packt Publishing.

Jerry Lee Ford, Jr. 2011. "*Ruby Programming*".

Hartl, Michael. 2011. *Ruby on Rails 3 Tutorial*. Pearson Education, Inc. Boston, United State.

Peter Cooper. 2010. *Beginning Ruby: From Novice to Professional, Second Edition (Beginning from Novice to Professional)*. Apress.

Prabowo Pudjo Widodo, Herlawati. 2011. *Menggunakan UML (Unified Modeling Language)*. Informatika Bandung.

Yuni Sugiarti. 2013. "*Analisis & Perancangan UML (Unified Modeling Language)*". Graha Ilmu. Yogyakarta.