

# PENGEMBANGAN PERANGKAT LUNAK E-COMMERCE BERORIENTASI ASPEK MENGUNAKAN USE CASES

Muhammad Aris Ganiardi <sup>\*1</sup>

<sup>\*1</sup>Jurusan Manajemen Informatika, Politeknik Negeri Sriwijaya, Jalan Srijaya Negara Bukit Besar Palembang 30139, E-mail : marisg2010@gmail.com

## *Abstrak*

Perangkat lunak berorientasi aspek merupakan perangkat lunak yang memiliki kualitas mudah digunakan ulang (reusability) dan diperbaiki (maintainability) yang lebih baik dibandingkan dengan perangkat lunak berorientasi objek. Pada pengembangan perangkat lunak berorientasi aspek concern-concern yang tersebar di objek-objek dipisahkan dan dimodularisasi menjadi sebuah aspek. Untuk memudahkan identifikasi dan pemisahan concern maka digunakan Use Cases pada fase analisis dan perancangan pada siklus pengembangan perangkat lunak. Uses Cases digunakan untuk mengidentifikasi dan memisahkan concern karena Use Cases menggambarkan interaksi langsung antara pengguna dan sistem. Studi kasus yang digunakan dalam peneliti adalah aplikasi web e-commerce pada mahasiswa dan alumni wirausaha Inkubator Bisnis Polsri. Hasil yang dihasilkan pada penelitian ini diharapkan dapat membantu mahasiswa dan alumni wirausaha Inkubator Bisnis Polsri untuk memasarkan produk mereka dan memudahkan pembeli untuk memesan sebuah produk di Inkubator Bisnis Polsri.

**Kata kunci :** *Pengembangan Perangkat Lunak Berorientasi Aspek, Use Cases, dan e-Commerce*

## 1. PENDAHULUAN

Paradigma pengembangan perangkat lunak yang paling populer saat ini adalah pengembangan perangkat lunak berorientasi objek. Perangkat lunak yang dikembangkan dengan menggunakan paradigma berorientasi objek dikonstruksi dari objek-objek yang didapat dari fase analisis siklus pengembangan perangkat lunak. Sebuah objek didapatkan berdasarkan modularisasi atribut-atribut dan operasi-operasi yang terjadi di sebuah perangkat lunak. Keuntungan penggunaan paradigm ini adalah kemudahan penggunaan ulang objek-objek pada perangkat lunak lain dan mudah dilakukan perbaikan objek pada perangkat lunak.

Kepopuleran paradigma berorientasi objek tidak membuat paradigma ini mempunyai kelemahan. Kelemahan pertama adalah pada paradigma ini, objek dimodularisasi berdasarkan sekumpulan data. Efek dari modularisasi berdasarkan sekumpulan data adalah muncul satu fungsi yang digunakan di banyak objek (kode *tangling*) dan satu objek digunakan di banyak fungsi (kode *scattering*) (Clarke,2005). Fungsi dan objek ini disebut dengan *concern*. Kelemahan yang kedua paradigma berorientasi objek adalah proses pembangunan perangkat lunak lebih menekankan pada faktor fungsionalitas dibandingkan dengan faktor non-fungsionalitasnya. Padahal dalam implementasinya kadang kala faktor non-fungsionalitas seperti keamanan lebih memiliki peranan dibandingkan dengan faktor fungsionalitas. Akibat kelemahan tersebut objek-objek akan saling bergantung satu dengan yang lain. Apabila ada satu kesalahan di sebuah objek maka kesalahan tersebut akan mempengaruhi objek yang lain. Secara global kesalahan-kesalahan di atas akan mempengaruhi kualitas perangkat lunak.

Kelemahan-kelemahan pengembangan perangkat lunak berorientasi objek diperbaiki oleh sebuah paradigma baru yang disebut pengembangan perangkat berorientasi aspek. Perangkat lunak berorientasi aspek merupakan perangkat lunak yang dibangun dengan cara memisahkan penyebaran *concern* (kode *tangling* dan *scattering*) yang terdapat di berbagai objek. Struktur kode program perangkat lunak mudah dipahami dan berisikan *method- method* inti bisnis utama yang dimiliki oleh objek-objek. *Concern-concern* yang tersebar dipisahkan dan dimodularisasi menjadi aspek yang bersifat mudah digunakan ulang. Perangkat lunak yang dibangun menggunakan paradigma berorientasi aspek lebih mudah untuk dirawat untuk ditambah atau dikurangi fiturnya.

Siklus pengembangan perangkat lunak berorientasi aspek sama seperti siklus pengembangan perangkat lunak lainnya. Secara umum fase-fase yang akan dilakukan adalah analisis, perancangan, pengkodean, pengujian, dan pemeliharaan perangkat lunak. Pada pengembangan perangkat lunak berorientasi aspek ada satu fase tambahan lagi yaitu fase identifikasi dan pemisahan *concern* menjadi aspek yang akan digunakan dalam perangkat lunak. Permasalahan yang muncul pada pengembangan perangkat lunak berorientasi aspek ini adalah “Bagaimana cara mengidentifikasi dan memisahkan *concern* menjadi aspek perangkat lunak pada fase analisis dan digambarkan pada fase perancangan ? Untuk menyelesaikan permasalahan tersebut peneliti menggunakan *Use Cases* untuk mengidentifikasi dan memisahkan *concern* menjadi aspek pada fase analisis, digambarkan pada fase perancangan, dan dimplementasikan pada fase pengkodean.

## 2. METODOLOGI PENELITIAN

### Metode Pengumpulan Data dan Kebutuhan

#### a. Studi Literatur

Metode ini dilakukan dengan mengkaji beberapa literatur yang berkaitan dengan penelitian pengembangan perangkat lunak berorientasi aspek, pemrograman berorientasi aspek. Literatur-literatur tersebut diperoleh dari :

- Buku-buku dan jurnal-jurnal penelitian baik oleh penulis dari dalam negeri maupun dari luar negeri.
- Data Perguruan Tinggi Negeri dan Swasta yang ada di kota Palembang.
- Informasi dari media masa, seperti surat kabar dan internet.

#### b. Survei

Pelaksanaan metode ini dengan melakukan observasi, kuesioner dan wawancara langsung pada para mahasiswa dan alumni wirausaha yang tergabung dalam Inkubator Bisnis Politeknik Negeri Sriwijaya Palembang. Peneliti mendatangi langsung ke tempat usaha para mahasiswa dan alumni wirausaha tersebut yang tersebar di berbagai wilayah kota Palembang, melakukan pengamatan pada proses bisnis mereka dan wawancara berkaitan dengan kebutuhan mereka terhadap perangkat lunak *e-commerce* dalam transaksi *on-line* dengan para pembeli.

### Pembangunan perangkat lunak *e-commerce* berorientasi aspek

Metode pengembangan perangkat lunak *e-commerce* yang akan digunakan pada penelitian ini adalah metode spiral. Secara umum metode spiral memiliki lima fase yaitu : (1) Analisis, (2) perancangan, (3) Pengkodean, (4) Pengujian, (5) Pemeliharaan. Metode spiral merupakan pengembangan dari metode spiral. Metode ini dipilih dalam pengembangan perangkat lunak karena metode ini mengizinkan pengembang kembali ke fase sebelumnya ketika ditemukan sebuah kesalahan. Sebagai contoh jika saat ini pengembang berada di fase pengkodean dan pengembang menemukan kesalahan terjadi di fase analisis maka pengembang dapat kembali ke fase analisis. Pada fase analisis tersebut pengembang dapat memperbaiki kesalahan tersebut sampai benar. Perangkat lunak ini nantinya akan dibuat dengan menggunakan bahasa pemrograman Java menggunakan IDE (*Integrated Development Environment*) Eclipse dan DBMS (*Database Management System*) MySQL di atas Sistem

Operasi Windows 7 (*seven*). Pemilihan bahasa pemrograman Java, Eclipse dan MySQL dilakukan karena bersifat *freeware* kecuali untuk sistem operasi Windows 7 (*seven*).

Berikut ini penjelasan tahapan pengembangan perangkat lunak menggunakan metode spiral

a. Analisis

1. *Pengumpulan kebutuhan*: data yang dikumpulkan adalah kebutuhan primer dan sekunder. Kebutuhan primer berasal dari data yang diambil langsung dari para *user* dan *stakeholder* pengguna perangkat lunak *e-commerce* seperti pembeli, penjual, dan pakar di bidang ilmu pengetahuan dan kebutuhan sekunder berasal dari data buku dan internet. Tahap ini sering disebut juga dengan tahapan akuisisi kebutuhan perangkat lunak. Teknik yang digunakan untuk mengumpulkan kebutuhan adalah wawancara, observasi, kuesioner, dan dokumentasi.
2. *Analisis kebutuhan*: Kebutuhan yang telah terkumpul diinterpretasi maksud dari kebutuhan yang didapat. Kebutuhan-kebutuhan yang berasal dari *user* dan *stakeholder* dianalisis untuk menentukan prioritas implementasinya. Interpretasi sebuah kebutuhan dapat berupa data, batasan, aturan, skenario, dan fungsionalitas perangkat lunak.
3. *Analisis basis data*: kasus yang telah terkumpul kemudian dibentuk sesuai dengan format basis data supaya bisa dimasukkan ke dalam basis data seperti pembuatan tabel kasus, pembuatan kunci, relasi antar tabel dan pembentukan *query*.
4. *Analisis model aplikasi*: model aplikasi yang akan digunakan berbasis *web* yang nantinya dapat didistribusikan secara melalui internet.
5. *Analisis perangkat lunak*: perangkat lunak yang digunakan adalah bahasa pemrograman java, IDE Eclipse, DBMS MySQL serta Sistem Operasi Windows 7 (*seven*).
6. *Analisis perangkat keras*: Analisa kebutuhan perangkat keras terhadap sistem seperti kecepatan *processor*, kapasitas memori utama dan memori sekunder.

b. Perancangan

1. *Rancangan basis data*: Rancangan basis data merupakan lanjutan dari analisa basis data. Perancangan basis data dilakukan dengan menggunakan DBMS MySQL serta melakukan pembuatan *query-query* yang nanti akan digunakan oleh sistem.
2. *Rancangan antarmuka*: Merancang tampilan masukan dan keluaran yang berbasis GUI (*Graphical User Interface*) menggunakan IDE Eclipse.
3. *Rancangan Objek* : Merancang modul-modul program dalam bentuk objek yang nantinya akan digunakan pada saat pengkodean sistem. Rancangan modul dapat berbentuk algoritma, notasi UML dan *pseudo-code*.
4. *Rancangan Aspek* : Mengidentifikasi cross cutting concern yang terdapat pada objek-objek program dan memisahkannya menjadi sebuah aspek pada program.

c. Pengkodean

1. *Pembuatan kode modul basis data*: kode modul basis data dibuat terpisah dengan kode sistem sehingga lebih bersifat *reusable*. Kode modul basis data berisi operasi basis data seperti membuat koneksi ke basis data, *insert*, *update*, *delete* dan *query*.
2. *Pembuatan kode komponen program*: kode program dibuat dan dimodularisasi menjadi objek berdasarkan spesifikasi yang telah ditetapkan dari hasil analisis.
3. *Pembuatan Aspek*: cross cutting concern yang terdapat pada objek dimodularisasi menjadi sebuah aspek.

d. Pengujian

1. *Pengujian basis data*: pengujian koneksi basis data dan akurasi *query* basis data.
2. *Pengujian sistem*: pengujian secara keseluruhan dari sistem baik dari masukan, proses dan keluaran sistem.
3. *Pengujian program*: pengujian ini dilakukan untuk mengetahui kualitas aspek perangkat lunak.

### 3. HASIL DAN PEMBAHASAN

#### 1. Kebutuhan Fungsional dan Kebutuhan Non Fungsional

##### Kebutuhan Fungsional

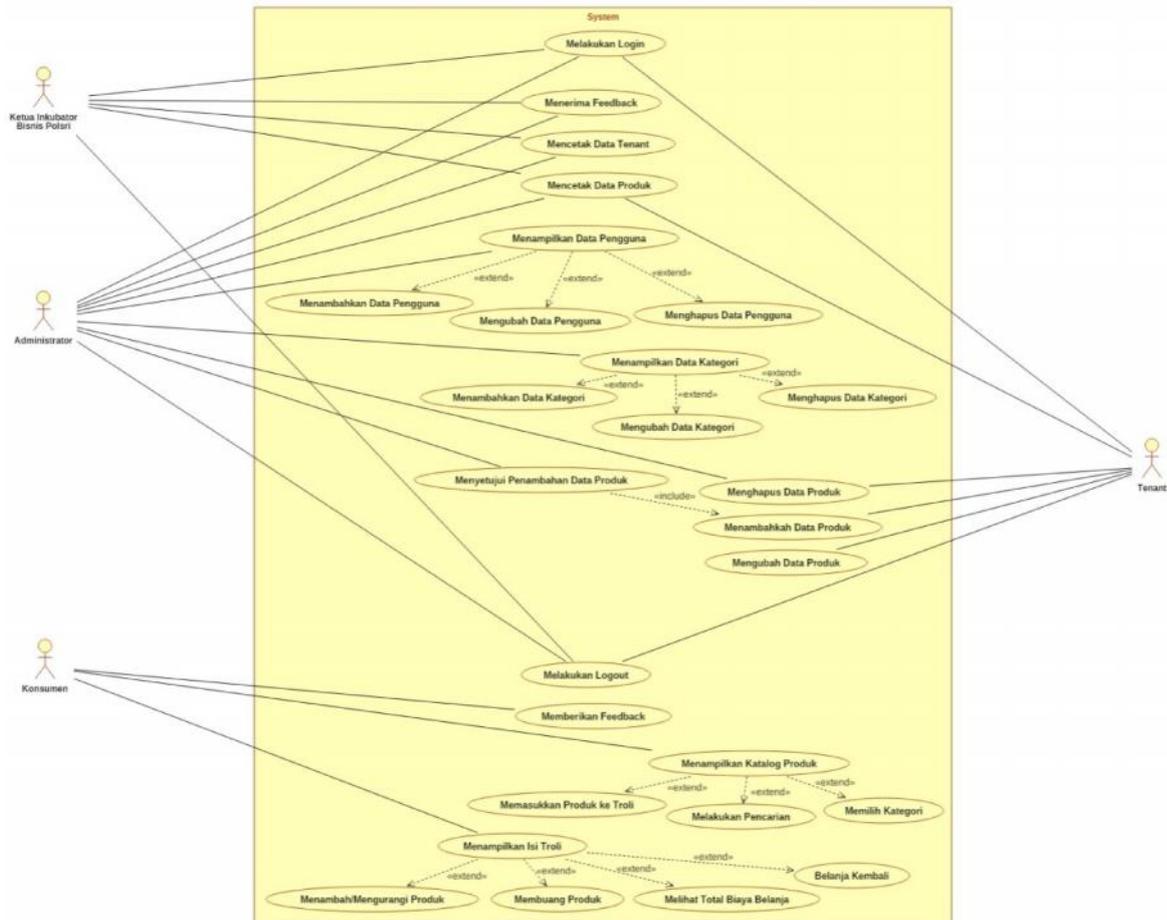
- a. Sistem mampu melakukan proses otentifikasi yaitu proses keamanan yang akan memvalidasi pengguna pada saat memasuki sistem serta mengecek hak akses yang dimiliki si pengguna berdasarkan akun yang mereka gunakan.
- b. Sistem mampu melakukan pengelolaan data pengguna baik berupa menambahkan, mengubah, atau pun menghapus data pengguna.
- c. Sistem mampu melakukan pengelolaan data kategori baik berupa menambahkan, mengubah, atau pun menghapus data tenant.
- d. Sistem mampu melakukan pengelolaan data produk baik berupa menambahkan, mengubah, atau pun menghapus data produk.
- e. Sistem mampu mencetak data-data di dalam basis data baik berupa data tenant dan produk secara kolektif maupun per tenant.
- f. Sistem mampu menyetujui setiap penambahan data produk.
- g. Sistem mampu melakukan proses mengeluarkan pengguna dari sistem (*logout*).
- h. Sistem mampu menampilkan katalog produk dengan fitur memasukkan produk-produk yang dibeli ke dalam troli, melakukan pemilihan kategori atau pun pencarian produk.
- i. Sistem mampu melakukan pengolahan troli baik berupa menambah/mengurangi produk, membuang produk, kembali berbelanja serta menampilkan total biaya belanja berdasarkan isi troli.
- j. Sistem mampu melakukan pemberian dan penerimaan *feedback*.

##### Kebutuhan non fungsional

Memiliki halaman antarmuka yang *friendly* berupa:

- a. Tampilan dapat langsung dimengerti oleh pengguna awam.
- b. Desain yang dibuat dan disusun sedemikian rupa agar bagian-bagian mudah dibedakan serta penggunaan animasi yang tidak terlalu berlebihan.
- c. Penggunaan simbol *hyperlink* berdasarkan standarisasi/notasi simbol yang umum sehingga mudah diketahui.

**Diagram Use Cases**



Gambar 1 Diagram Use Case

**Definisi Aktor**

Berikut adalah deskripsi pendefinisian aktor pada perangkat lunak *e-commerce*:

Tabel 1. Definisi Aktor

No.	ID	Aktor	Deskripsi
1.	AK-001	Ketua Inkubator Bisnis Polsri	Orang yang bertugas dan memiliki hak akses untuk melakukan pengontrolan atau pemantauan terhadap aktivitas yang terjadi pada sistem.
			Ketua Inkubator Bisnis Polsri dapat melakukan ototentifikasi atau <i>login</i> untuk mendapatkan hak aksesnya dan <i>logout</i> . Adapun hak akses yang dimiliki seorang Ketua Inkubator Bisnis Polsri pada sistem adalah menerima feedback; memantau data produk dan data tenant

			yang ada secara kolektif serta dapat melakukan pencetakan data.
2.	AK-002	Administrator	Administrator dapat berupa staf atau tenaga pelaksana pada organisasi yang ditunjuk oleh ketua dengan hak akses berupa ototentifikasi atau <i>login</i> dan <i>logout</i> ; mengelola data pengguna yaitu menambahkan, mengubah, dan menghapus data pengguna; mengelola data kategori yaitu menambahkan, mengubah, dan menghapus data kategori; menyetujui penambahan data produk; dan menerima <i>feedback</i> . Seorang administrator juga memiliki hak akses yang sama dengan ketua dalam hal memantau serta mencetak data produk dan data tenant.
3.	AK-003	Tenant	Tenant adalah orang yang memproduksi serta menjual produk-produk mereka di Inkubator Bisnis Politeknik Negeri Sriwijaya. Seorang tenant memiliki hak akses untuk melakukan ototentifikasi atau <i>login</i> dan <i>logout</i> ; pengelolaan data produk yaitu menghapus, menambahkan, dan mengubah data produk. Untuk penambahan data produk memerlukan persetujuan administrator.
4.	AK-004	Konsumen	Konsumen adalah orang yang diperbolehkan untuk melihat-lihat katalog produk yang dijual di aplikasi tanpa harus melakukan proses ototentifikasi; melakukan pencarian produk; memilih kategori; memasukkan produk ke troli; melihat isi troli diaman dapat menambah/mengurangi kuantitas produk, membuang produk, kembali berbelanja, dan melihat total biaya belanja.

## 1. Definisi Use Case

Berikut adalah deskripsi pendefinisian *use case* pada perangkat lunak *e-commerce*:

Tabel 5.2. Definisi Use Case

No.	ID	Use Case	Deskripsi
1.	UC-001	Melakukan <i>Login</i>	Merupakan proses untuk melakukan otentifikasi atau <i>login</i> pada saat pengguna akan masuk ke aplikasi. Proses <i>login</i> ini menentukan hak akses atau otorisasi apa saja yang dimiliki pengguna.
2.	UC-002	Menerima <i>Feedback</i>	Merupakan proses untuk menampilkan kesan dan pesan ( <i>feedback</i> ) yang konsumen telah berikan melalui aplikasi.
3.	UC-003	Mencetak Data Tenant	Merupakan proses dimana menampilkan data tenant untuk dicetak.
4.	UC-004	Mencetak Data Produk	Merupakan proses untuk melihat dan mencetak data-data produk yang terdaftar pada tabel dalam aplikasi. <i>Use case</i> ini merupakan pencetakan terhadap data produk secara kolektif.
5.	UC-005	Menampilkan Data Pengguna	Merupakan proses menampilkan data pengguna yang telah tersimpan di basis data sistem dengan meliputi 3 (tiga) buah <i>extend</i> yaitu menambahkan data pengguna, mengubah data pengguna, dan menghapus data pengguna.
6.	UC-006	Menambahkan Data Pengguna	Merupakan proses memasukkan data pengguna baru ke dalam basis data
7.	UC-007	Mengubah Data Pengguna	Merupakan proses mengubah data pengguna yang ada di basis data.
8.	UC-008	Menghapus Data Pengguna	Merupakan proses menghapus data pengguna dari basis data.
9.	UC-009	Menampilkan Data Kategori	Merupakan proses menampilkan data kategori yang ada. Terdiri dari 3 (tiga) buah <i>extend</i> yaitu menambahkan kategori, mengubah kategori, dan menghapus kategori.
10.	UC-010	Menambahkan Data Kategori	Merupakan proses memasukkan data kategori baru ke dalam basis data.
11.	UC-011	Mengubah Data Kategori	Merupakan proses mengubah data kategori yang ada di basis data.

12.	UC-012	Menghapus Data Kategori	Merupakan proses menghapus data kategori dari basis data.
13.	UC-013	Menghapus Data Produk	Merupakan proses menghapus data produk dari basis data.
14.	UC-014	Menambahkan Data Produk	Merupakan proses menambahkan data produk ke dalam basis data.
15.	UC-015	Mengubah Data Produk	Merupakan proses mengubah data produk yang ada di basis data.
16.	UC-016	Menyetujui Penambahan Data Produk	Merupakan proses untuk menyetujui penambahan data produk yang dilakukan oleh tenant untuk dipajang pada halaman aplikasi.
17.	UC-017	Melakukan <i>Logout</i>	Merupakan proses untuk melakukan <i>logout</i> atau keluar dari sistem.
18.	UC-018	Memberikan <i>Feedback</i>	Merupakan proses untuk memberikan kesan dan pesan ( <i>feedback</i> ).
19.	UC-019	Menampilkan Katalog Produk	Merupakan proses menampilkan data produk-produk yang ada di basis data dan telah disetujui oleh administrator. Terdiri dari 3 (tiga) buah <i>extend</i> yaitu memasukkan produk ke troli, melakukan pencarian, dan memilih kategori.
20.	UC-020	Memasukkan Produk ke Troli	Merupakan proses mengumpulkan produk-produk yang telah dibeli untuk dihitung total biaya.
21.	UC-021	Melakukan Pencarian	Merupakan proses mencari produk atau lainnya berdasarkan kata kunci yang dimasukkan ke dalam kotak isian.
22.	UC-022	Memilih Kategori	Merupakan proses memilih kategori untuk menampilkan produk-produk tertentu berdasarkan kategori yang dipilih.
23.	UC-023	Menampilkan Isi Troli	Merupakan proses menampilkan isi troli yaitu produk-produk apa saja yang telah dimasukkan ke dalam troli. Proses ini memiliki 4 (empat) buah <i>use case</i> sebagai ekstensi yaitu menambah/mengurangi produk, membuang produk, melihat total biaya dan belanja kembali.
24.	UC-024	Menambah/ Mengurangi Produk	Merupakan proses untuk menambah atau pun mengurangi kuantitas produk yang ada di dalam troli.

25.	UC-025	Membuang Produk	Merupakan proses membuang produk yang telah dimasukkan ke dalam troli.
26.	UC-026	Melihat Total Biaya Belanja	Merupakan proses menampilkan hasil perhitungan total biaya atas produk-produk yang telah dimasukkan ke dalam troli.
27.	UC-027	Belanja Kembali	Merupakan proses untuk melihat atau memilih produk-produk lainnya yang ada di katalog produk.

### Aspek Perangkat Lunak E-Commerce

Berdasarkan penggunaan *use cases* dalam identifikasi aspek pada analisis perangkat lunak *e-commerce*, ada sembilan aspek yang berhasil diidentifikasi dan dimodularisasi. Dari kesembilan aspek tersebut ada dua aspek yang tidak diintegrasikan ke perangkat lunak *e-commerce*, sedangkan tujuh aspek lainnya diintegrasikan. Aspek-aspek yang dimodularisasi dibagi menjadi empat kategori yaitu aspek *pooling* yang berfungsi untuk mengatur penggunaan sumber daya basisdata, aspek *validator* yang berfungsi untuk mevalidasi data, aspek *security* berfungsi untuk keamanan perangkat lunak *e-commerce*, dan aspek *testing* berfungsi untuk menguji secara internal kode program perangkat lunak *e-commerce*. Daftar aspek perangkat lunak *e-commerce* secara lengkap ditunjukkan pada tabel 2.

Tabel 2. Daftar Aspek perangkat lunak *e-commerce*

No.	Kategori	Aspek E-Commerce	Keterangan
1	Aspek <i>Pooling</i>	Database	Diimplementasikan
		ConnectionChecking	Diimplementasikan
2	Aspek <i>Validator</i>	ValidateString	Diimplementasikan
		ValidateNumeric	Diimplementasikan
		ValidateNull	Diimplementasikan
		Errors	Tidak diimplementasikan
		Status	Tidak diimplementasikan
3	Aspek <i>Security</i>	GeneratePassword	Diimplementasikan
4	Aspek <i>Testing</i>	Tracer	Diimplementasikan

Penjelasan aspek-aspek perangkat lunak *e-commerce* adalah sebagai berikut :

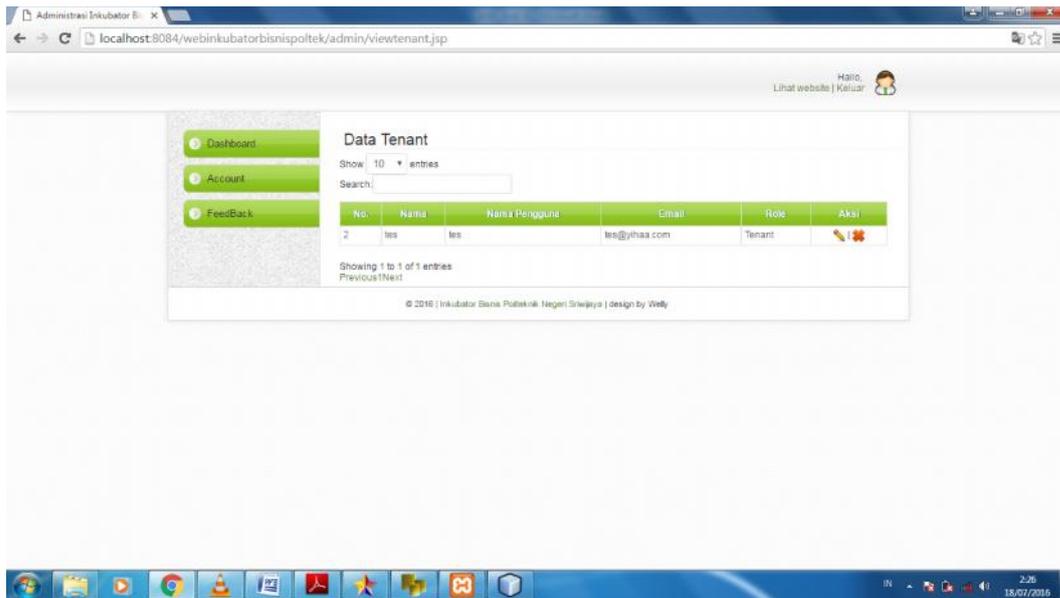
1. Aspek Database adalah aspek perangkat lunak *e-commerce* yang berfungsi untuk koneksi antara perangkat lunak *e-commerce* dengan basisdata commerce. Aspek ini digunakan untuk menyimpan data perangkat lunak *e-commerce* dan menampilkan data tersebut di halaman *web* perangkat lunak *e-commerce*.
2. Aspek ConnectionChecking adalah aspek perangkat lunak *e-commerce* yang berfungsi sebagai pendeteksi hubungan antara perangkat lunak *e-commerce* dengan basisdata commerce.
3. Aspek ValidateString adalah aspek yang berfungsi untuk mevalidasi *input*-an dalam bentuk huruf yang dimasukkan oleh pengguna. perangkat lunak *e-commerce* hanya mengijinkan data yang dimasukkan dalam bentuk huruf pada *field-field* tertentu pada sebuah *form*. Jika pengguna memasukan data *input*-an dalam bentuk angka maka perangkat lunak *e-commerce* tidak memproses dan menyimpan data *input*-an tersebut ke dalam basisdata.
4. Aspek ValidateNumeric adalah aspek yang berfungsi untuk mevalidasi *input*-an dalam bentuk angka yang dimasukkan oleh pengguna. perangkat lunak *e-commerce* hanya

mengijinkan data yang dimasukkan dalam bentuk angka pada *field-field* tertentu pada sebuah *form*. Jika pengguna memasukan *input-an* dalam bentuk huruf maka perangkat lunak *e-commerce* tidak memproses dan menyimpan *input-an* tersebut ke dalam basisdata.

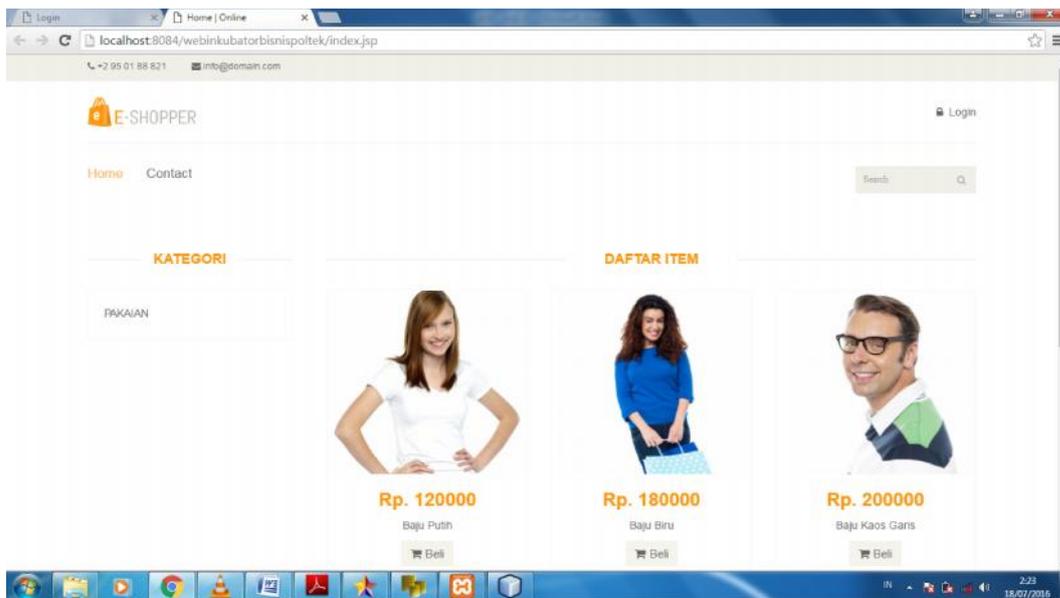
5. Aspek *ValidateNull* adalah aspek yang berfungsi untuk mevalidasi *input-an* dalam bentuk “**kosong**” yang dimasukkan oleh pengguna. Pada *field-field* form tertentu, perangkat lunak *e-commerce* mewajibkan pengguna memasukkan data. Data *field-field* yang harus dimasukkan adalah data-data berbentuk kunci primer seperti NIM mahasiswa, NIP dosen atau data karakteristik unik entitas FUOLC. Jika pengguna memasukan *input-an* dalam bentuk kosong maka FUOLC akan menolak memproses dan menyimpan *input-an* tersebut ke dalam basisdata.
6. Aspek *Errors* adalah aspek yang berfungsi untuk menampilkan pesan kesalahan jika adalah kesalahan *input-an* yang dimasukkan oleh pengguna. Aspek ini tidak diimplementasikan dan diintegrasikan ke perangkat lunak *e-commerce* karena keterbatasan teknik implementasi pemrograman berorientasi aspek. Pada aspek *Errors* harus mendeklarasikan objek *errors* yang bertipe data *Hashtable* sebagai parameternya. Ketika diintegrasikan, aspek *Errors* tidak mendeteksi *join point* yang berasal dari *class*. Secara teknis harus ada kesesuaian antara parameter *join point* aspek dan *method class*.
7. Aspek *Status* adalah aspek yang berfungsi untuk menampilkan pesan aksi-aksi yang dilakukan oleh pengguna. Contoh pesan seperti : “pesan data sudah tersimpan di basisdata”, “pesan data sudah diubah”, dan “pesan data sudah dihapus di basisdata”. Aspek ini tidak diimplementasikan dan diintegrasikan di perangkat lunak *e-commerce* karena bentuk *method* status yang akan diubah menjadi aspek *Status* sudah dalam bentuk primitif. Jika diimplementasikan maka bentuk aspek dan *method* di *class* sama.
8. Aspek *GeneratePassword* adalah aspek yang berfungsi untuk mengenkripsi *password* pengguna perangkat lunak *e-commerce*. Untuk mencegah pengguna lain yang tidak memiliki hak akses mengetahui *password* pengguna maka setiap *password* pengguna yang masuk ke perangkat lunak *e-commerce* akan dienkripsi.
9. Aspek *Tracer* adalah aspek yang berfungsi untuk menelusuri aliran kerja *method-method* program perangkat lunak *e-commerce* saat sebuah *class* dieksekusi. Aspek ini digunakan untuk menguji struktur internal perangkat lunak *e-commerce*. Hasil pengujian dengan menggunakan aspek *Tracer* akan didapatkan urutan *method-method* yang dieksekusi saat sebuah fungsionalitas mengeksekusi sebuah *class*.

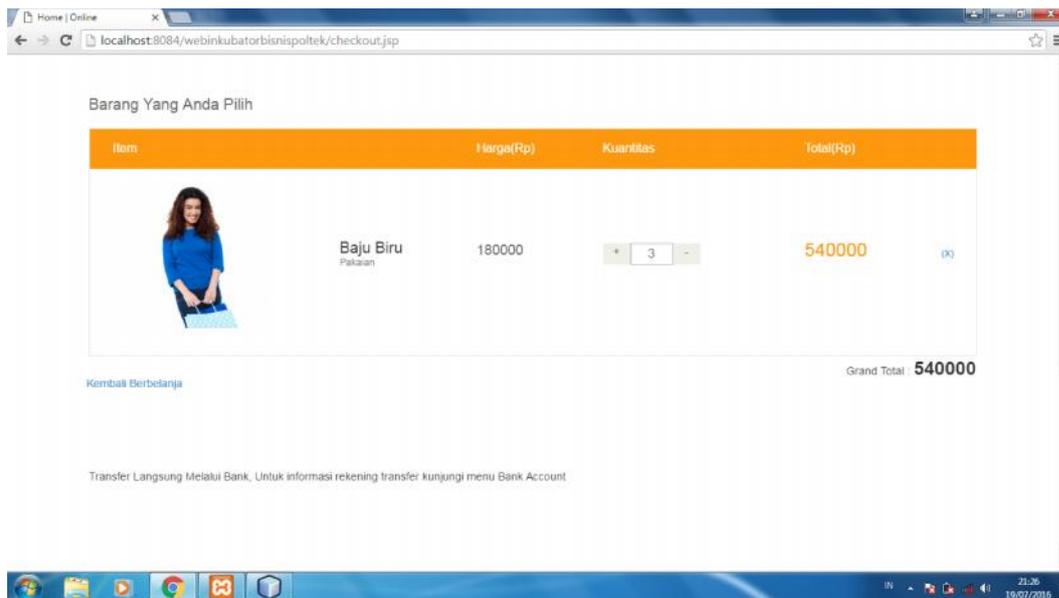
### **Tampilan perangkat lunak *e-commerce***

Gambar 2 dan 3 menampilkan halaman-halaman web yang ada perangkat lunak *e-commerce*. Halaman-halaman web ini tidak terpengaruh paradigman yang digunakan untuk membuat perangkat lunak *e-commerce*.



Gambar 2. Form pengolahan data tenant





Gambar 3. Keranjang perangkat lunak

#### 4. KESIMPULAN

1. Perangkat lunak *e-commerce* ini memberikan banyak keuntungan bagi segala pihak yang terlibat di dalam proses bisnis Inkubator Bisnis Politeknik Negeri Sriwijaya. Salah satunya adalah semakin mudahnya para tenant dalam menjangkau pangsa pasar yang secara otomatis akan meningkatkan omzet penjualan.
2. Perangkat lunak *e-commerce* berorientasi aspek yang dikembangkan memiliki karakteristik independen yang sangat tinggi. Tidak ada lagi *concern-concern* yang sama tersebar diberbagai class-class perangkat lunak. Karakteristik ini memudahkan adaptasi perangkat lunak e-commerce terhadap perubahan-perubahan yang berasal dari pengguna.
3. Use cases digunakan untuk mengidentifikasi penyebaran concern-concern perangkat lunak berdasarkan spesifikasi perangkat lunak. Spesifikasi perangkat lunak merupakan hasil konversi kebutuhan-kebutuhan pengguna. Keunggulan menggunakan use case, karena use cases menggambarkan langsung interaksi antara pengguna dengan sistem.

#### 5. SARAN

Perangkat lunak *e-commerce* berorientasi aspek masih berupa purwarupa yang harus diuji kualitasnya. Pengujian dilakukan berdasarkan kualitas perangkat lunak yaitu faktor kualitas maintainability. Faktor kualitas ini dipilih untuk mengetahui tingkat pengaruh perubahan spesifikasi perangkat lunak terhadap program komputernya. Diharapkan pada penelitian selanjutnya dilakukan pengujian terhadap perangkat lunak *e-commerce* berorientasi aspek ini.

#### UCAPAN TERIMAKASIH

Teriring salam dan doa, kami sampaikan ucapan terimakasih kepada beberapa pihak yang telah membantu dalam publikasi jurnal ini, terutama redaksi jurnal Jupiter Politeknik Negeri Sriwijaya.

DAFTAR PUSTAKA

- Clarke .2005. *Aspect-Oriented Analysis and Design: The Theme Approach*. Addison Wesley. USA.
- Clarke .2004. Theme: An Approach for Aspect-Oriented Analysis and Design. ICSE 2004. Proceedings 26th International Conference on . IEEE.
- Filman. 2004. *Aspect-Oriented Software Development*. Addison Wesley. USA.
- Hidehiko Masuhara<sup>1</sup> and Gregor Kiczales. 2003. *Modeling Crosscutting in Aspect-Oriented Mechanisms*. Proceeding ECOOP. Pp 2-28.
- Howard Podeswa. (2009). *The Business Analyst's Handbook*. Course Technology. USA.
- Jacobson dan Pan Wei Ng. 2004. *Aspect-Oriented Software Development with Use Cases*. Addison Wesley. USA.
- Jing Zhang. 2007. *Aspect Composition in The Motorola Aspect-Oriented Modeling Weaver*. Journal Of Object Technology. Vol. 6. No. 7. Special Issue: Aspect-Oriented Modeling.
- Joseph D. Gradecki. 2003. *Mastering AspectJ Aspect-Oriented Programming in Java*. Wiley Publishing .Inc. Canada.
- Lidia Fuentes. 2007. *Designing And Weaving Aspect-Oriented Executable UML Models*. Journal Of Object Technology. Vol. 6. No. 7. Special Issue: Aspect-Oriented Modeling.
- Ramnivas Laddad. 2006. *AspectJ in Action Practical Aspect-Oriented Programming*. Manning Publication. USA.
- Roger S Pressman. 2001. *Software Engineering: A Practitioner's Approach (5<sup>th</sup> Ed.)*. Mc Graw-Hill. USA.
- Russell Miles. 2005. *AspectJ Cookbook*. O'Reilly. USA.
- Sommerville. 2006. *Software Engineering (8<sup>th</sup> Ed)*. Addison Wesley. USA.
- Tudor B. Ionescu et al. 2010. "An Aspect-Oriented Approach for the Development of Complex Simulation Software". Journal Of Object Technology. Vol. 9. No. 1.
- Turban, E., King, D., Lee, J., Warkentin, M., Chung, H.M. 2002 .*Electronic Commerce 2002: A Managerial Perspective*,Prentice Hall, New Jersey, USA.
- Victor Pankratius. 2005. *Aspect-Oriented Re-engineering of E-learning Courseware*. The Learning Organization Vol. 12 No. 5.