

TUGAS AKHIR

APLIKASI MOBILE COMMERCE MEMANFAATKAN WEB SERVICE PADA BLACKBERRY



Disusun Oleh :

Nama : Dani Rachmat Triyadi

NIM : 04 523 266

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2012

LEMBAR PENGESAHAN PEMBIMBING

**APLIKASI MOBILE COMMERCE MEMANFAATKAN WEB
SERVICE PADA BLACKBERRY**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Disusun Oleh :

Nama : Dani Rachmat Triyadi

NIM : 04 523 266

Yogyakarta, Februari 2012

Dosen Pembimbing,

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

LEMBAR PENGESAHAN PENGUJI

**APLIKASI MOBILE COMMERCE MEMANFAATKAN WEB
SERVICE PADA BLACKBERRY**

Disusun Oleh :

Nama : Dani Rachmat Triyadi

NIM : 04 523 266

**Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Informatika**

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta , Februari 2012

Tim Penguji

R.Teduh Dirgahayu, ST., M.Sc., Ph.D.

Syarif Hidayat ,S.Kom., M.IT

Affan Mahtarani, S.Kom, M.T

Tanda Tangan



Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri

Univesitas Islam Indonesia



(Yudi Prayudi, S.Si, M.Kom)

PERSEMBAHAN

Tugas Akhir ini Saya Persembahkan Kepada :

Allah SWT Rabb Semesta Alam

*“Sesungguhnya sholatku , ibadahku, Hidupku dan Matiku hanyalah untuk Allah
SWT Semesta Alam”*

(Al-Qur’an Al An-am 162)

Kekasih Allah, Kanjeng Nabi Muhammad SAW

“Nabi lebih utama bagi orang mukmin dari pada mereka sendiri...”

(Al-Qur’an Al Ahzab 6)

Papa dan Mamaku

*“Terima Kasih atas semua doa dan restunya, kasih sayang dan cintanya,
pengorbanan dan perhatiaannya, kepercayaan san segala sukungannya”*

Kakak-kakaku, Doddy dan Ruddy atas Supportnya

MOTTO

“ Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain “

(Q.S Alam Nasyrh ayat 6 dan 7)

“.....Sesungguhnya Allah tidak merubah keadaan suatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri.....”

“Dintengah kesulitan terdapat kesempatan”

“Itu tidak berarti saya cerdas, itu karena hanya saya tetap dengan masalah tersebut lebih lama”

“Ingatlah dibalik kesulitan ada kemudahan yang menanti”

“Kegagalan adalah guru yang terbaik dan kegagalan adalah kesuksesan yang tertunda asalkan kita tidak menyerah dengan kegagalan itu”

KATA PENGANTAR



Assalamu'alaikum wr.wb.

Dengan mengucapkan puji syukur kehadirat Allah SWT, serta limpahan rahmatNya, tak lupa penyusun mengucapkan syukur alhamdulillah sehingga laporan tugas akhir telah diselesaikan dengan baik. Sholawat dan salam kepada junjungan Nabi Muhammad SAW dengan sya'faatnya kami mengharapakan keselamatan baik keselamatan baik didunia maupun di akhirat.

Tugas akhir ini, merupakan salah satu syaratakademik yang harus ditempuh seorang mahasiswa untuk memperoleh gelar sarjana di Jurusan Teknik Informatika, Fakultas teknologi Industri, Universitas Islam Indonesia.

Dalam pelaksanaan dan pengerjaan akhirini, penyusun mendapatkan banyak bantuan dari berbagai pihak. Oleh karena itu penyusun mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT yang telah memberikan segala sesuatu yang terbaik untukku.
2. Bapak Yudi Prayudi, S.Si, M.kom selaku ketua jurusan Teknik Informatika Fakultas, Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku dosen pembimbing tugas akhir yang telah memberikan pengarahan dan bimbingan selama kerja praktek dan penulisan laporan ini.
4. Kedua orang tua (Soeryadi dan Siti Khodijah) dan kakak-kakaku (Doddy dan Ruddy) tercinta, yang telah memberikan doa, dukungan dan motivasi untuk menyelesaikan tugas akhir ini.

5. Teman-teman Jurusan Informatika dan teman-teman yang tidak dapat disebutkan satu-persatu, terima kasih atas dorongan, bantuan dan motivasinya.

Penyusun menyadari bahwa laporan ini masih jauh dari sempurna, karena keterbatasan kemampuan dan pengalaman. Oleh karena itu penulis mengharapkan kritik dan saran demi perbaikan-perbaikanyang digunakan sebagai koreksi kekurangan yang terdahulu, dimasa mendatang.

Akhir kata penyusun berharap semoga laporan tugas akhir ini dapatmendatangkan manfaat bagi penyusun dan dapat menjadi refrensi dikemudian hari bagi para pembaca serta mendatangkan ridha dari Allah SWT.Amin

Wassalamu'alaikum wr.wb

Yogyakarta, Februari 2012



Penyusun.

SARI

Saat ini Blackberry sudahlah menjadi sebagian kebutuhan hidup dari berbagai macam kalangan, mulai dari kalangan bawah sampai kalangan menengah ke atas. Kemampuan layanan *push e-mail*, telepon, sms, Menjelajah internet, dan berbagai kemampuan nirkabel lainnya. Penggunaan gadget canggih ini begitu fenomenal belakangan ini, sampai menjadi suatu kebutuhan untuk *fashion*.

Dengan demikian blackberry dapat dijadikan sebagai peluang usaha dalam bisnis perdagangan di dunia maya, tentunya akan membuka peluang baru untuk menarik konsumen dan juga dapat dijadikan sebuah media promosi. Salah satu contoh yang dapat diterapkan adalah pembuatan aplikasi penjualan seperti Barbequa, yang dapat diakses menggunakan perangkat *mobileblackberry*.

Hadirnya teknologi *Web service* sebagai metode yang ditujukan untuk menjembatani akses antara *mobile device* dengan sistem penjualan *online* melalui jaringan internet, sehingga dapat mewujudkan kemudahan dalam berbelanja tersebut, dengan membuat Aplikasi *Mobile Commerce* memanfaatkan *Web Service* pada Blackberry.

Kata kunci : *Blackberry, Mobile Commerce, Web Service*.

TAKARIR

<i>Web Service</i>	Perkembangan arsitektur teknologi yang memungkinkan untuk menerapkan konsep integrasi data pada jaringan internet berbasis web
<i>E-commerce</i>	Pemasaran barang dan jasa melalui sistem elektronik seperti internet, atau jaringan komputer.
<i>M-commerce</i>	Proses perdagangan secara <i>online</i> melalui perangkat-perangkat nirkabel seperti <i>pager</i> , PDA dan telepon seluler
<i>Mobile Device</i>	Sebuah peralatan komputer seukuran kantong, yang memiliki layar tampilan dengan masukan sentuhan atau papan ketik mini
<i>API</i>	Sekumpulan perintah, fungsi dan protokol yang dapat digunakan untuk membangun perangkat lunak
<i>Checkout</i>	Proses transaksi online yang ditujukan untuk menyederhanakan proses pembayaran saat belanja

DAFTAR ISI

HALAMAN JUDUL	I
HALAMAN PENGESAHAN PEMBIMBING.....	II
HALAMAN PENGESAHAN PENGUJI	III
HALAMAN PERSEMBAHAN.....	IV
MOTTO	V
KATA PENGANTAR.....	VI
SARI	VIII
TAKARIR.....	IX
DAFTAR ISI.....	X
DAFTAR TABEL.....	XV
DAFTAR GAMBAR.....	XVI
BAB I.....	1
PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 RUMUSAN MASALAH.....	2
1.3 BATASAN MASALAH.....	2
1.4 TUJUAN TUGAS AKHIR	3
1.5 MANFAAT TUGAS AKHIR.....	3
1.6 METODOLOGI PENELITIAN.....	3
1.6.1 Metode Pengumpulan Data.....	3
1.6.2 Metodologi Pengembangan Sistem.....	4
1.7 SISTEMATIKA PENULISAN	5
BAB II	6
LANDASAN TEORI.....	6
2.1 MOBILE COMMERCE.....	6

2.2	BLACKBERRY.....	6
2.2.1	<i>Sejarah Blackberry.....</i>	6
2.2.2	<i>Blackberry sebagai Client.....</i>	7
2.3	TEKNOLOGI WEB SERVICE	7
2.3.1	<i>Sejarah Perkembangan Web Service</i>	7
2.3.2	<i>Keuntungan Web Service</i>	9
2.3.3	<i>Komponen Pendukung Web Service.....</i>	10
2.4	JARINGAN KOMPUTER	16
2.4.1	<i>Konsep Client-Server</i>	16
2.4.2	<i>Protokol Jaringan</i>	17
2.4.3	<i>HTTP (Hyper Text Transfer Protocol).....</i>	18
2.4.4	<i>Internet.....</i>	18
BAB III.....		19
METODOLOGI.....		19
3.1	ANALISIS KEBUTUHAN SISTEM.....	19
3.3.1	<i>Aplikasi Web (server).....</i>	19
3.3.1.1	Zend Framework	19
3.3.2	<i>Aplikasi Mobile (client).....</i>	19
3.3.2.1	Java SE JDK Versi 6	19
3.3.2.2	Eclipse IDE	20
3.3.2.3	Blackberry JDE Plugin for Eclipse	21
3.3.3	<i>Aplikasi Web service (client-server)</i>	21
3.3.3.1	JSON (JavaScript Object Notation)	21
3.3.3.2	SOAP (Simple Object Access Protocol)	21
3.3.4	<i>Library (pustaka)</i>	22
3.3.4.1	PHP SOAP	22
3.3.4.2	kSOAP.....	22
3.3.5	<i>Tools Pengembangan (Mozilla Firefox)</i>	22
3.3.5.1	JSONView.....	23
3.3.5.2	SOA Client	23

3.4 PERANCANGAN APLIKASI <i>MOBILE COMMERCE</i>	23
3.4.1 <i>Perancangan Sistem Web service</i>	24
3.4.1.1 <i>Sistem Web service</i>	24
Sistem.....	24
3.4.1.2 <i>Format Web service</i>	24
3.4.1.3 <i>Aktifitas Web service</i>	25
3.4.2 <i>Perancangan Sistem Aplikasi</i>	27
3.4.2.1 <i>Use Case Diagram</i>	27
3.4.2.2 <i>Class Diagram</i>	29
3.4.2.3 <i>Sequence Diagram</i>	32
3.4.2.4 <i>Activity Diagram</i>	33
3.4.3 <i>Perancangan Masukan Sistem</i>	36
3.4.3.1 <i>Registrasi</i>	36
3.4.3.2 <i>Login</i>	36
3.4.3.3 <i>Pemesanan produk</i>	36
3.4.3.4 <i>Konfirmasi Pesanan</i>	36
3.4.3.5 <i>Perubahan Data Member</i>	36
3.4.4 <i>Perancangan Database</i>	37
3.4.4.1 <i>MySQL</i>	37
3.4.4.2 <i>BlackBerry Persistent Store</i>	42
3.4.5 <i>Perancangan Tampilan</i>	42
3.4.5.1 <i>Rancangan Antar Muka User</i>	42
3.4.5.2 <i>Rancangan antar Muka Admin</i>	44
BAB IV	46
IMPLEMENTASI	46
4.1 IMPLEMENTASI SISTEM	46
4.1.1 <i>Batasan Implementasi</i>	46
4.1.2 <i>Implementasi Web service</i>	46
4.1.2.1 <i>Web service JSON</i>	46
4.1.2.2 <i>Web service SOAP</i>	50

4.2	PENGUJIAN SISTEM.....	54
4.2.1	<i>Lingkungan Pengujian</i>	54
4.2.2	<i>Pengujian Sistem User</i>	54
4.2.2.1	Halaman Menu Utama (Home).....	54
4.2.2.2	Halaman Daftar Produk.....	55
4.2.2.3	Halaman Detail Produk.....	55
4.2.2.4	Halaman Keranjang Belanja	55
4.2.2.5	Halaman <i>Checkout</i> Akun	55
4.2.2.6	Halaman <i>Checkout</i> Registrasi	56
4.2.2.7	Halaman <i>Checkout</i> sebagai Anggota	56
4.2.2.8	Halaman <i>Checkout</i> Selesai.....	56
4.2.2.9	Halaman Akun Anggota.....	56
4.2.2.10	Halaman Konfirmasi Pesanan Anggota	57
4.2.2.11	Halaman Edit Profil Anggota.....	57
4.2.2.12	Halaman Testimonials.....	57
4.2.2.13	Halaman Bantuan	57
4.2.3	<i>Analisis Sistem</i>	58
4.2.3.1	Pengujian Kesalahan Deteksi Jaringan	58
4.2.3.2	Pengujian Kesalahan Penambahan Produk	58
4.2.3.3	Pengujian Kesalahan Proses Registrasi.....	58
4.2.4	<i>Pengujian Sistem terhadap Admin</i>	59
4.2.4.1	Halaman Menu Admin.....	59
4.2.4.2	Halaman Kelola Produk	59
4.2.4.3	Halaman Pesanan <i>User</i>	60
4.3	HASIL IMPLEMENTASI SISTEM.....	60
4.3.1	Kelebihan Aplikasi.....	60
4.3.2	Kekurangan Aplikasi.....	61
BAB V	62
PENUTUP	62
5.1	KESIMPULAN	62

5.2 SARAN.....	63
DAFTAR PUSTAKA.....	64
LAMPIRAN.....	65



DAFTAR TABEL

Tabel 3.1 Tabel categories	37
Tabel 3.2 Tabel products.....	37
Tabel 3.3 Tabel products_images	38
Tabel 3.4 Tabel products_comments	38
Tabel 3.5 Tabel users	39
Tabel 3.6 Tabel users_profile.....	39
Tabel 3.7 Tabel users_shipping.....	39
Tabel 3.8 Tabel orders.....	40
Tabel 3.9 Tabel orders_profile.....	40
Tabel 3.10 Tabel orders_detail.....	40
Tabel 3.11 Tabel options.....	41
Tabel 3.12 Tabel options_banks	41
Tabel 3.13 Tabel cities	41
Tabel 3.14 Tabel provinces.....	42
Tabel 4.1 Service Soap.....	52
Tabel 4.2 Keterangan tag WSDL.....	54

DAFTAR GAMBAR

Gambar 3.1 Arsitektur sistem	23
Gambar 3.2 Bagan Alur <i>Web service</i>	24
Gambar 3.3 Format <i>Web service</i>	25
Gambar 3.4 Aktifitas <i>Web service</i>	26
Gambar 3.5 Use Case Diagram	29
Gambar 3.6 Class Diagram	31
Gambar 3.7 Sequence Diagram.....	33
Gambar 3.8.1 Activity Diagram User	34
Gambar 3.8.2 Activity Diagram Pembelian User	35
Gambar 3.9 Rancangan antar muka <i>user</i>	43
Gambar 3.10 Rancangan antar muka admin	45
Gambar 4.1 JSON Daftar Produk.....	65
Gambar 4.2 JSON Detail Produk.....	65
Gambar 4.3 JSON Tarif	66
Gambar 4.4.1 JSON Informasi User	66
Gambar 4.4.2 Exception JSON Informasi <i>user</i>	67
Gambar 4.5.1 JSON Pemesanan <i>User</i>	67
Gambar 4.5.2 Exception JSON Pemesanan <i>User</i>	67
Gambar 4.6.1 Skema Dokumen WSDL.....	53
Gambar 4.6.3 SOAP WSDL	68
Gambar 4.6.3 Service Method dengan SOA Client	68
GAMBAR PENGUJIAN SISTEM USER	69
Gambar 4.7 Halaman Utama.....	69
Gambar 4.8 Halaman Daftar Produk.....	69
Gambar 4.9 Halaman Detail Produk	70
Gambar 4.10 Halaman Keranjang Belanja.....	70
Gambar 4.11 Halaman <i>Checkout</i> Akun	71
Gambar 4.12 Halaman <i>Checkout</i> Registrasi	71
Gambar 4.13.1 Halaman <i>Checkout</i> Login Anggota.....	72

Gambar 4.13.2 Halaman <i>Checkout</i> sebagai Anggota.....	72
Gambar 4.14 Halaman <i>Checkout</i> Selesai.....	73
Gambar 4.15.1 Halaman Akun Login Anggota	74
Gambar 4.15.2 Halaman Akun Anggota.....	74
Gambar 4.16 Halaman Konfirmasi Pembayaran Pesanan Anggota.....	75
Gambar 4.17 Halaman Edit Profil Anggota	76
Gambar 4.18 Halaman Testimonials.....	76
Gambar 4.19 Halaman Bantuan	77
GAMBAR ANALISIS SISTEM	78
Gambar 4.20 Halaman Deteksi Jaringan.....	78
Gambar 4.21 Halaman Kesalahan Penambahan Produk.....	78
Gambar 4.22 Halaman Kesalahan Proses Registrasi	79
GAMBAR PENGUJIAN SISTEM ADMIN	80
Gambar 4.23.1 Halaman Menu Admin pada Web.....	80
Gambar 4.23.2 Halaman Menu Admin pada <i>Mobile</i>	80
Gambar 4.24.1 Halaman Menu Tambah Produk pada <i>Mobile</i>	81
Gambar 4.24.2 Halaman Menu Edit Produk pada <i>Mobile</i>	81
Gambar 4.25.1 Halaman Menu Lihat Pesanan <i>User</i> pada <i>Mobile</i>	82
Gambar 4.25.2 Halaman Menu Ubah Pesanan <i>User</i> pada <i>Mobile</i>	83
Gambar 4.25.3 Halaman Lihat Pesanan <i>User</i> pada <i>WebServer</i>	83
Gambar 4.25.4 Halaman Ubah Pesanan <i>User</i> pada <i>WebServer</i>	84

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi internet memungkinkan untuk melakukan pemasaran barang dan jasa secara *online*, atau yang biasa disebut *e-commerce*. Secara umum *e-commerce* menggunakan perangkat PC untuk mendorong transformasi bisnis, sebagai pengguna dapat melakukan transaksi tanpa dibatasi dimensi waktu (masih dibatasi kebebasan lokasi), maka *mobile commerce* mencoba menjadi solusi untuk menyediakan tingkat pelayanan yang lebih baik dalam memfasilitasi transaksi dengan menghilangkan keterbatasan yang terkait dengan lokasi. Segala bentuk transaksi penjualan dan pembelian produk yang dilakukan melalui perangkat *mobile*, sehingga transaksi bisa dilakukan kapan saja dan dimana saja.

Inovasi teknologi nirkabel (*wireless*) yang berkembang pesat, menghadirkan teknologi *smartphone* (telepon pintar) sebagai telepon genggam yang menyajikan teknologi tinggi menyerupai komputer, salah satunya adalah Blackberry. Kemampuan *gadget* ini dalam komunikasi data melalui media internet, diperlukan dalam pengembangan *mobile commerce* untuk melakukan akses terhadap sistem penjualan *online* melalui jaringan internet.

Berkembangannya teknologi komputasi terdistribusi (*distributed computing*) yang memungkinkan dilakukannya komputasi pada banyak mesin dan hasilnya dapat dimanfaatkan oleh banyak mesin pula. Salah satu teknologi komputasi terdistribusi yang saat ini sedang berkembang adalah teknologi *Web Service*.

Web service adalah sebuah sistem perangkat lunak yang dirancang untuk mendukung inter-operasi dalam interaksi mesin ke mesin melalui sebuah jaringan. Interaksi dilakukan melalui mekanisme atau protokol tertentu. *Web service* menggunakan protokol internet seperti HTTP, SMTP, dan FTP sebagai protokol transportasi data yang ditujukan untuk menjembatani akses antara *mobile device* dengan aplikasi web (sistem penjualan *online*). Sehingga mempermudah proses pertukaran data tanpa adanya perbedaan perangkat keras, sistem operasi, dan bahasa pemrograman (lintas platform).

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan masalah sebagai berikut :

Membangun aplikasi *mobile commerce* pada Blackberry untuk dapat melakukan pertukaran data dari sebuah sistem penjualan *online*, dengan memanfaatkan *web service*.

1.3 Batasan Masalah

Berdasarkan permasalahan yang sudah diuraikan di atas, perlu diadakan pembatasan terhadap penganalisaan rencana pembuatan aplikasi, agar perancangan aplikasi dapat di desain dengan efisien dan efektif.

Batasan masalah dari rencana pembuatan aplikasi ini adalah sebagai berikut :

1. Studi kasus aplikasi ini adalah penjualan sepatu online.
2. Sistem aplikasi ini semi *M-Commerce*, dimana proses transaksi pembayaran dilakukan diluar secara manual melalui transfer bank.
3. Disini proses pengiriman barang tidak dibahas secara detail.

1.4 Tujuan Tugas Akhir

Tujuan dari tugas akhir ini adalah merancang dan mengimplementasikan *mobile commerce* sebagai media pembelian sepatu sehingga dapat memudahkan proses pembelian.

1.5 Manfaat Tugas Akhir

1. Menyajikan suatu sistem pembelian sepatu *online* dengan menyediakan fasilitas-fasilitas yang mudah dipahami oleh konsumen/pengguna.
2. Sistem aplikasi ini dapat dipergunakan oleh para pembeli secara efektif karena dapat dilakukan dimana saja dan kapan saja.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan pengembangan sistem.

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data terdiri dari tiga cara yaitu :

1. Wawancara

Wawancara dilakukan pada beberapa konsumen yang akan membeli sepatu secara online. Dari wawancara ini diperoleh pendapat mereka terhadap permasalahan yang ada, mengenai faktor lokasi yang membuat mereka kesulitan untuk membeli produk.

2. Observasi

Observasi dilakukan melalui pengamatan dan pencatatan data terhadap data-data produk. Hal ini bertujuan untuk mendapatkan data yang akan digunakan dalam pembuatan sistem.

3. Studi Pustaka

Cara ini dilakukan dengan mengumpulkan data dari buku, internet dan *literature* lain yang berhubungan dengan *mobile commerce* untuk menambah informasi serta referensi dalam menyelesaikan tugas akhir ini.

1.6.2 Metodologi Pengembangan Sistem

Dalam metodologi pengembangan sistem meliputi:

1. Analisis sistem

Dalam tahap ini dilakukan pengamatan terhadap permasalahan proses pembelian produk yang kemudian digunakan sebagai dasar dalam pembuatan sistem ini. Analisis yang dilakukan pada input, output dan operasi pada sistem.

2. Perancangan Sistem

Dalam perancangan sistem ini digunakan UML (Unified Modeling Language) untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Termasuk di dalamnya adalah perancangan database yang akan menyimpan data yang dibutuhkan sistem sesuai dengan perancangannya.

3. Implementasi Sistem

Implementasi sistem dilakukan dengan cara pembuatan sistem sesuai dengan perancangan yang telah dibuat sebelumnya.

4. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibuat guna mengetahui normal tidaknya sistem berjalan.

1.7 Sistematika Penulisan

Sistematika penulisan terdiri atas lima bab, dengan susunan sebagai berikut :

Bab I Pendahuluan

Pada bab ini berisi tentang latar belakang, perumusan masalah dan pembatasan masalah dan sistematika penulisan yang digunakan untuk penyusunan skripsi ini.

Bab II Landasan Teori

Pada bab ini berisi mengenai *mobile-commerce*, sejarah blackberry, teknologi *web service*, jaringan komputer serta membahas perangkat lunak yang digunakan untuk membangun aplikasi ini.

Bab III Metodologi

Pada bab ini membahas perancangan dan pembuatan aplikasi.

Bab IV Pengujian Aplikasi

Pada bab ini akan dilakukan pengujian terhadap hasil akhir aplikasi.

Bab V Penutup

Pada bab ini akan dibuat kesimpulan dan saran-saran yang diperlukan untuk pengembangan aplikasi lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Mobile Commerce

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadipemutusan atau terputusnya komunikasi. Sedangkan *commerce* adalah perdagangan, pemesanan, transaksi atau jual beli.

Jadi, *mobile commerce* adalah proses perdagangan secara *online* melalui perangkat-perangkat nirkabel seperti *pager*, PDA dan telepon seluler. *M-commerce* (*mobile commerce*) dikenal sebagai generasi berikutnya dari *e-commerce*. *M-commerce* adalah segala bentuk transaksi penjualan dan pembelian produk yang dilakukan melalui perangkat bergerak seperti telepon seluler atau telepon genggam, sehingga transaksi bisa dilakukan kapan saja dan dimana saja.

2.2 Blackberry

Blackberry adalah perangkat seluler yang memiliki kemampuan layanan *push e-mail*, telepon, sms, Menjelajah Internet, dan berbagai kemampuan nirkabel lainnya. Penggunaan gadget canggih ini begitu fenomenal belakangan ini, sampai menjadi suatu kebutuhan untuk *fashion*. BlackBerry pertama kali diperkenalkan pada tahun 1997 oleh perusahaan Kanada, Research In Motion (RIM). Kemampuannya menyampaikan informasi melalui jaringan data nirkabel dari layanan perusahaan telepon genggam hingga mengejutkan dunia.

2.2.1 Sejarah Blackberry

Blackberry pertama kali diperkenalkan di Indonesia pada pertengahan Desember 2004 oleh operator Indosat dan perusahaan Starhub. Perusahaan Starhub merupakan pengejawantahan dari RIM yang merupakan rekan utama BlackBerry. Pasar BlackBerry kemudian diramaikan oleh dua operator besar

lainnya di tanah air yakni Excelcom dan Telkomsel. Akibat tuntutan pemerintah Indonesia, Blackberry akhirnya membuka kantor perwakilan di Indonesia pada November 2010

2.2.2 Blackberry sebagai Client

Penggunaan gadget blackberry yang canggih begitu fenomenal belakangan ini, dengan banyaknya pengguna blackberry di Indonesia, dan menjadi suatu kebutuhan untuk gaya hidup zaman sekarang. Dari segi tersebut dijadikanlah Blackberry sebagai perangkat mobile yang digunakan untuk aplikasi *M-Commerce* ini. Oleh karena itu, dengan menggunakan aplikasi mobile-commerce pada blackberry tentunya akan membuka peluang baru untuk menarik konsumen dan juga dapat dijadikan sebuah media promosi.

2.3 Teknologi Web Service

Web service merupakan fenomena yang sangat panas saat ini karena, banyak kelebihan yang ditawarkan oleh Web Service terutama interoperabilitas tinggi dan penggunaannya yang dapat diakses kapanpun dan dimanapun selama mesin kita terhubung oleh jaringan internet salah satunya.

2.3.1 Sejarah Perkembangan Web Service

Perkembangan internet ternyata mampu mempengaruhi proses evolusi perkembangan sistem terdistribusi (*distributed system*), dari sistem yang berdasarkan pada platform proprietary (DCOM, CORBA, RMI) menuju platform yang lebih terbuka yaitu web. Momentum perkembangan ini diawali dengan kehadiran XML (*eXtensible Markup Language*) untuk mendukung pertukaran data berbasis web secara lebih fleksibel. Keberadaan standard XML kemudian ikut mendorong penetapan standard protokol komunikasi untuk mendukung mekanisme transfer data pada sistem terdistribusi melalui jaringan internet yang dikenal sebagai Simple Object Access Protocol (SOAP) pada tahun 1998. Pada akhirnya kedua teknologi ini (XML dan SOAP) kemudian mendasari perkembangan arsitektur teknologi yang memungkinkan untuk menerapkan

konsep integrasi data pada jaringan internet berbasis web, yang kemudian dikenal sebagai *web service*. World Wide Web Consortium (W3C) mendefinisikan *web service* sebagai “ sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML melalui protokol internet”.

Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler. *Web service* bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi firewall.

Web service dapat diimplementasikan baik untuk lingkungan internal (intranet) maupun untuk lingkungan publik (internet). Dalam lingkungan internal, *web service* memberikan kemudahan dan solusi baru untuk mengintegrasikan berbagai platform aplikasi yang ada dalam organisasi atau perusahaan. Pada lingkungan publik, *web service* akan memperbaharui konsep transaksi data berbasis internet, dan menawarkan teknologi yang mampu mengintegrasikan data

dari berbagai sumber data dalam organisasi atau perusahaan terkait maupun dengan partner bisnis secara lebih mudah, sederhana dan murah.

Dan secara umum, *web service* akan mengubah paradigma para pengguna maupun pengembang perangkat lunak, dari berbasis API (*application program interface*) menjadi berbasis pesan (*message-based*).

2.3.2 Keuntungan Web Service

Beberapa keuntungan menggunakan web service, sebagai berikut :

1. Lintas Platform

Penggunaan *web service* memungkinkan komputer-komputer yang berbeda sistem operasi data saling bertukar data. Untuk mengakses web service, sistem operasi yang anda gunakan tidak perlu sama dengan sistem operasi yang digunakan oleh *web service* tersebut. Yang perlu diketahui hanyalah method apa saja yang disediakan *web service* tersebut.

2. Language Independent

Sebuah *web service* dapat diakses menggunakan bahasa pemrograman apa saja. Web service yang dibuat dengan bahasa pemrograman PHP dapat diakses oleh berbagai bahasa pemrograman mulai dari bahasa pemrograman PHP itu sendiri, JSP, Delphi, VB.NET, bahkan memungkinkan untuk diakses oleh mobile device seperti handphone dengan memanfaatkan aplikasi berbasis Java untuk mobile device atau J2ME.

3. Jembatan Penghubung dengan database

Web service dapat dijadikan “jembatan” penghubung antar aplikasi dengan database. Sehingga aplikasi tidak lagi memerlukan driver database dan tidak perlu mengetahui database apa yang digunakan oleh server serta bagaimana struktur database tersebut. Aplikasi tersebut cukup mengetahui method atau fungsi yang disediakan *web service* untuk memanfaatkan fasilitasnya.

4. Mempermudah Proses Pertukaran Data

Penggunaan *web service* dapat mempermudah dan mempercepat pertukaran data diantara mereka daripada mereka harus menyesuaikan aplikasi dari database yang digunakan, yang tentu akan memakan waktu, tenaga, dan biaya yang besar.

2.3.3Komponen Pendukung Web Service

Web service memiliki komponen pendukung yang merupakan dasar terbentuknya *web service*. Beberapa komponen pendukung *web service* pada aplikasi ini sebagai berikut :

1. *Exentensible Markup Languange* (XML)

XML termasuk kedalam keluarga Markup Language dan masih satu keluarga dengan HTML (Hypertext Markup Language).XML merupakan dasar penting atas terbentuknya web service.Web service dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilnya dengan menggunakan XML.Karena XML berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protocol HTTP.

Dan yang terpenting XML bersifat platform independent sehingga informasi di dalamnya dibaca oleh aplikasi dengan platform apapun selama aplikasi tersebut mampu menerjemahkan tag-tag XML.Berikut ini adalah contoh sebuah dokumen XML :

```
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
```

```
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

XML terletak pada inti *Web Service*, yang digunakan untuk mendeskripsikan data. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan partner luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antara satu dengan yang lain.

2. *Simple Object Access Protocol (SOAP)*

SOAP mulanya merupakan kepanjangan dari *Simple Object Access Protocol*. Namun seiring dengan perkembangan *Web Service*, kepanjangan tersebut tidak lagi dipakai. Dalam *Web Service*, SOAP merupakan format standar dokumen berbentuk XML yang digunakan untuk melakukan proses request dan response antara *Web Service* dengan aplikasi yang memanggilnya. Peran SOAP di dalam teknologi *Web Service* adalah sebagai protokol pemaketan untuk pesan-pesan yang digunakan secara bersama.

SOAP menjadi sangat mudah diterima oleh berbagai pihak, terutama oleh kalangan TI karena protokol ini memanfaatkan berbagai teknologi yang sudah ada dan telah banyak digunakan. Misalnya untuk protokol transport, yang paling sering digunakan adalah HTTP, walaupun dimungkinkan untuk menggunakan protokol transport lainnya. Sedangkan untuk format data, digunakan XML yang tidak diragukan lagi manfaat dan perannya di dalam pertukaran data.

SOAP menspesifikasikan secara jelas bagaimana cara untuk meng-encode header HTTP dan file XML sehingga program pada suatu komputer dapat

memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan.

Sebuah pesan SOAP terdiri dari SOAP Envelop dan bisa terdiri dari Attachments atau tidak memiliki attachment. SOAP Envelop tersusun dari SOAP Header dan SOAP Body. SOAP Attachment membolehkan data non-XML untuk dimasukkan ke dalam pesan SOAP

Sebuah pesan SOAP adalah sebuah dokumen XML yang berisi elemen-elemen berikut.

- a. Elemen *envelope* mengidentifikasi dokumen XML sebagai sebuah pesan SOAP.
 - b. Elemen *header* berisi informasi yang sifatnya tambahan dan opsional.
 - c. Elemen *body* berisi panggilan dan respon terhadap informasi yang dipertukarkan antara *Web Service* dan aplikasi.
 - d. Elemen *fault* berisi pesan kesalahan yang terjadi pada waktu proses. Elemen ini bersifat opsional.
3. *Web Service Description Language* (WSDL)

Dokumen dalam format XML yang isinya menjelaskan informasi detail dari sebuah web service. Di dalam WSDL dijelaskan method-method apa saja yang tersedia dalam *web service*, parameter apa saja yang diperlukan untuk memanggil sebuah method, dan apa hasil atau tipe data yang dikembalikan oleh method yang dipanggil tersebut.

Dalam dokumen WSDL terdapat pernyataan versi XML yang dipakai. Kemudian sebuah komponen root Definition yang merupakan tag tertinggi dalam dokumen WSDL. Tag-tag lainnya harus berada di dalam tag ini. Di dalamnya terdapat berbagai atribut berupa sebuah Name (nama service) dan sebuah TargetNameSpace dan nol atau lebih namespaces. Namespaces digunakan untuk menghindari konflik penamaan ketika beberapa service atau aplikasi diintegrasikan. Berikut adalah contoh sebuah dokumen WSDL :

```

<definitions name="Soap_Service" targetNamespace="http://barbequa.nitelab.web.id/api/soap">
- <types>
- <xsd:schema targetNamespace="http://barbequa.nitelab.web.id/api/soap">
- <xsd:complexType name="ArrayOfProduct">
- <xsd:complexContent>
- <xsd:restriction base="soap-enc:Array">
  <xsd:attribute ref="soap-enc:arrayType" wsdl:arrayType="tns:Product[]"/>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
- <xsd:complexType name="Product">
- <xsd:all>
  <xsd:element name="id" type="xsd:string"/>
  <xsd:element name="product_name" type="xsd:string"/>
  <xsd:element name="price" type="xsd:string"/>
  <xsd:element name="img" type="xsd:string"/>
  <xsd:element name="comments" type="xsd:string"/>
  <xsd:element name="ratings" type="xsd:string"/>
  <xsd:element name="publish" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
- <xsd:complexType name="Order">
- <xsd:all>
  <xsd:element name="id" type="xsd:string"/>
  <xsd:element name="code" type="xsd:string"/>
  <xsd:element name="total" type="xsd:string"/>
  <xsd:element name="phone" type="xsd:string"/>
  <xsd:element name="transfer" type="xsd:string"/>
</xsd:all>
</xsd:complexType>

```

Sebuah komponen Definition mengandung komponen-komponen berikut:

- a. *Types*, mendefinisikan tipe data yang digunakan dalam pesan, misal XSD. Di dalam bagian ini didefinisikan tipe-tipe data kompleks yang digunakan oleh Web Service. Tipe data sederhana seperti integer, double, dan string tidak didefinisikan lagi.
- b. *Message*, merupakan bagian yang mengandung informasi yang dibutuhkan untuk melakukan operasi, serta parameter input dan output apa saja yang akan digunakan.
- c. *Operation* adalah sebuah deskripsi abstrak dari aksi yang dilakukan oleh service. Setiap operation dapat disamakan dengan method atau function

call dalam sebuah bahasa pemrograman tradisional. Di sini aksi SOAP didefinisikan dan pesan di-encode.

- d. *Port Type*, mendefinisikan operasi yang tersedia dalam Web Service dan pesan yang digunakan untuk melakukan operasi.
- e. *Binding*, menspesifikasikan Port Type, mendefinisikan operasi, protokol yang digunakan untuk memanggil operasi tersebut (HTTP atau SMTP), bagaimana operasi tersebut dilakukan, dan gaya binding yang diterapkan (document atau RPC).
- f. *Port*, mendefinisikan alamat jaringan Web Service. Bagian ini biasanya direpresentasikan dengan sebuah string url http sederhana.
- g. *Service*, dapat dianggap sebagai kontainer untuk sekumpulan fungsi sistem yang telah diekspos ke protokol-protokol berbasis web.

4. *JavaScript Object Notation (JSON)*

Suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca-manusia serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. Format JSON bertipe teks sehingga mudah untuk ditransportasikan menggunakan protocol HTTP.

Walaupun JSON didasarkan pada subset bahasa pemrograman JavaScript dan umumnya digunakan dengan bahasa tersebut, JSON dianggap sebagai format data yang tak tergantung pada suatu bahasa. Kode untuk pengolahan dan pembuatan data JSON telah tersedia untuk banyak jenis bahasa pemrograman, menjadikannya bersifat platform independent sehingga informasi di dalamnya dibaca oleh aplikasi dengan platform apapun selama aplikasi tersebut mampu menerjemahkan atau mem-parsing objek JSON.

Kemampuan untuk merepresentasikan struktur data sederhana dan larik asosiatif, dapat mendukung antarmuka *web service* yang beroperasi pada objek JSON sebagai data input dan format output. Web

serviceJSONdisediakandengankonsistenyang samadenganlayananutukSOAP, WSDLdanXML.

Penggunaan JSON sebagai komponen format web service didasarkan atas kemampuan dan kesederhanaanya.Disini bisa kita bisa membandingkan antara JSON dan XML berdasarkan kriteria berikut.

1. *Human readability*

Mudah dibaca adalah kreteria yang penting karena dalam proses pembuatan dan mendebug akan memudahkan programmer. Struktur JSON lebih mudah dibaca dan di debug dibandingkan denganXML.

XML

```
<person>
<firstname>Ibnu</firstname>
<lastname>daqiqil</lastname>
</person>
```

JSON

```
{
  "firstName" : "Ibnu",
  "lastName" : "Daqiqil"
};
```

2. Kecepatan

Dibandingkan dengan XML, parsing JSON membutuhkan waktu lebih cepat.

3. Format JSON sudah didukung

Di beberapa bahasa pemrograman telah mendukung format data JSON, seperti di PHP > 5.1.x, fungsi `json_encode()` dan `json_decode()`. Selain itu, di Java juga sudah tersedia library untuk pengaksesan JSON, sebagai contoh [Json-lib](#).

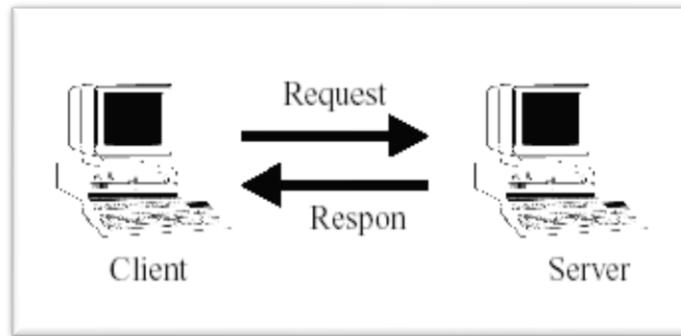
2.4 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan *hardware/software* yang terhubung dengan jaringan. Setiap komputer, printer atau *peripheral* yang terhubung dengan jaringan disebut *node*. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node.

2.4.1 Konsep Client-Server

Pada dasarnya, semua transaksi atau perpindahan data di jaringan komputer tidak terlepas dari konsep *client-server*. Perpindahan data ini berlangsung karena adanya permintaan (*request*) dari salah satu komputer ke komputer lain yang menyimpan data. Sebagai tanggapan permintaan data ini, maka komputer penyimpan data akan memberikan tanggapan (*response*).

Tanggapan ini berupa pengiriman data yang ingin diakses oleh komputer yang melakukan permintaan data. Dalam konsep *client-server*, komputer peminta data dinamakan sebagai *client* dan komputer pemilik data dinamakan sebagai *server*. Datanya sendiri dapat berupa antara lain *file*, *web*, *email* dan lain-lain. Implementasi dari konsep client server ini adalah program yang memiliki fungsi seperti dideskripsikan pada konsep tersebut. Contohnya sebuah program *web client* berfungsi mengajukan *request* berupa data *web*, sementara program yang berfungsi sebagai *web server* berfungsi menunggu permintaan dan mengirimkan data *web* kepada peminta data *web*.



Gambar 2.1 Konsep *Client-server*

2.4.2 Protokol Jaringan

Protokol adalah Suatu kesepakatan mengenai bagaimana komunikasi akan dilakukan. Dalam dunia komunikasi data komputer di dalam suatu jaringan, protokol mengatur bagaimana sebuah komputer berkomunikasi dengan komputer lain. Dalam jaringan komputer dapat digunakan banyak macam protokol tetapi agar dua atau lebih komputer dapat berkomunikasi, keduanya perlu menggunakan protokol yang sama. Protokol berfungsi mirip dengan bahasa.

Untuk mempermudah pengertian, penggunaan, desain serta agar terjadi penyeragaman di antara perusahaan pembuat peralatan jaringan komputer, *Internasional Standard Organization* (ISO) mengeluarkan suatu model lapisan jaringan yang disebut *Open Systems Interconnection* (OSI). Di dalam model OSI ini, proses pengolahan data dibagi dalam tujuh lapisan (*layer*) dimana masing-masing lapisan mempunyai fungsi sendiri-sendiri. Model OSI tidak membahas secara detail cara kerja dari tiap-tiap lapisannya.

Selain model OSI, ada juga model TCP/IP (*Transmission Control Protocol/Internet Protocol*) yang dikeluarkan oleh *Department of Defense America* (DOD). Jika OSI terdiri dari tujuh lapisan maka TCP/IP hanya terdiri dari empat lapisan. Komputer-komputer yang terhubung ke jaringan dapat saling berkomunikasi karena menggunakan protokol yang sama, yaitu protokol TCP/IP. Perbedaan jenis komputer dan sistem operasi tidak menjadi masalah. Komputer dengan sistem operasi *Windows* dapat

berkomunikasi dengan komputer *Macintosh* atau dengan *Sun SPARC* yang menjalankan *Solaris*.

2.4.3 HTTP (Hyper Text Transfer Protocol)

HTTP adalah suatu metode atau protokol untuk men-*downloadfile* ke komputer. Protokol ini berbasis *hyper text*, sebuah format teks yang umum digunakan di Internet.(Maselena, 2003).Sebuah alamat internet akan berawalan protokol ini. Sebagai contoh untuk mengakses situs *yahoo*, maka pada *URL (uniform resource locator)* atau alamat internet ditulis dengan *http://www.yahoo.com* Sedangkan halaman situs yang dibuka umumnya akan berupa *file HTML (Hyper Text Markup Language)*.

2.4.4 Internet

Internet adalah kepanjangan dari *Interconnection network*, merupakan interkoneksi antara komputer-komputer (*node*) di seluruh dunia yang membentuk sebuah jaringan komputer global. Internet adalah suatu istilah umum yang dipakai untuk menunjuk jaringan (*Network*) tingkat dunia yang terdiri dari komputer dan layanan atau servis untuk pemakai komputer, dan bermacam-macam sistem informasi termasuk *e-mail* (surat elektronik), *FTP (File Transfer Protocol)* dan *World Wide Web /www*.

BAB III

METODOLOGI

3.1 Analisis Kebutuhan Sistem

Dalam membangun aplikasi *mobile commerce* dibagi menjadi beberapa bagian yaitu aplikasi web (sebagai *server*), aplikasi mobile (sebagai *client*), aplikasi *client-server* (sebagai *middleware* yaitu *web service*), *library* dan web browser (sebagai *tools* pendukung). Aplikasi perangkat lunak tersebut dibuat dengan menggunakan bahasa pemrograman

3.3.1 Aplikasi Web (*server*)

Aplikasi ini adalah aplikasi berbasis web dengan menggunakan web *server* apache, *server-side* menggunakan Zend Framework PHP, dan *client-side* dengan bahasa markup HTML dan javascript. Aplikasi web dipanggil menggunakan web browser, seperti mozilla firefox, ie, chrome, dll. (*recommended* mozilla).

3.3.1.1 Zend Framework

Pada aplikasi web (*server*) pada sistem ini menggunakan Zend Framework. Zend Framework adalah sebuah framework open source yang digunakan untuk pengembangan aplikasi berbasis web dengan menggunakan PHP versi 5.

3.3.2 Aplikasi Mobile (*client*)

Dibutuhkan beberapa software yang digunakan untuk membuat aplikasi *client*. Diantaranya sebagai berikut :

3.3.2.1 Java SE JDK Versi 6

Di dalam Java SDK merupakan kumpulan tools yang berguna untuk membuat dan menjalankan program berbasis Java, didalamnya memiliki JRE dan JDK.

JRE (*Java Runtime Environment*) inilah yang memungkinkan sebuah program Java dapat berjalan di mesin Anda. JRE ini mengeksekusi binary-binary dari *class-class* dan mengirimnya ke JVM untuk diproses lagi ke prosesor. Setiap JRE pasti memiliki sebuah JVM di dalamnya untuk melakukan pemrosesan selanjutnya.

JDK (*Java Development Kit*) adalah semacam kotak peralatan (kit) yang digunakan untuk development. JDK ini berguna saat Anda menulis code program. Seperti halnya JRE, JDK juga memiliki JVM di dalamnya

JVM (*Java Virtual Machine*) merupakan jantung dari Java Platform. JVM ini adalah pihak yang bertanggung jawab untuk mengeksekusi program Java menjadi bahasa mesin untuk diproses oleh prosesor. JVM mampu menerjemahkan code-code Java ke hampir semua platform. JVM ini-lah yang membuat Java "*write once, run everywhere*" alias multi-platform.

3.3.2.2 Eclipse IDE

Sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent, target adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X*). Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*. Eclipse yang dipakai *Eclipse versi 3.4 (Ganymede) for Java Developers*.

3.3.2.3 Blackberry JDE Plugin for Eclipse

Research In Motion (RIM) meluncurkan Blackberry JDE (*Java Development Environment*) yang merupakan lingkungan dan program simulasi pengembangan yang benar-benar terintegrasi untuk membuat aplikasi Java Platform, Micro Edition (Java® ME) untuk smartphone BlackBerry® berbasis Java®.

BlackBerry JDE adalah lingkungan Java ME yang kompatibel dengan Mobile Information Device Profile (MIDP) untuk pengembang yang ingin aplikasi nirkabelnya benar-benar portabel. Selain itu, BlackBerry JDE menyediakan paket lengkap antarmuka dan utilitas untuk memanfaatkan beberapa fitur unik di smartphone BlackBerry.

Plugin BlackBerry_JDE_PluginFull_1.0.0.67. Versi plugin ini sudah *include* Blackberry JDK versi 4.5. Pada aplikasi ini ditambah Blackberry JDK versi 4.6 (*eJDE-4_6_0_19_Component*) untuk Blackberry OS versi 4.6 keatas (blackberry JDE support forward OS, dan tidak support *backward* OS).

3.3.3 Aplikasi Web service (client-server)

Aplikasi *web service (client-server)* sebagai pertukaran data antara *server* dan *client*. Pada aplikasi ini menggunakan menggunakan 2 jenis format *web service*, yaitu JSON dan SOAP.

3.3.3.1 JSON (JavaScript Object Notation)

Web service JSON digunakan untuk melakukan *read process*, karena formatnya teks sehingga lebih cepat dan ringan dalam *encoding* dari pada format XML. JSON sebagai alternatif dari XML yang lebih simple dan mudah dibaca oleh manusia (*more human-readable*).

3.3.3.2 SOAP (Simple Object Access Protocol)

Web service SOAP digunakan untuk melakukan *writeprocess*, memiliki setting atau parameter untuk mengirimkan data dari *client* ke *server*.

3.3.4 Library (pustaka)

Library atau pustaka adalah koleksi dari rutin-rutin program yang digunakan untuk membangun dan mengembangkan perangkat lunak. *Library* ini berisikan class-class yang digunakan sebagai bantuan untuk mengembangkan atau membuat sebuah *web service*. Berikut pustaka yang diperlukan:

3.3.4.1 PHP SOAP

Merupakan library extension dari PHP untuk menangani *web service* SOAP dengan PHP pada sisi *server*.

3.3.4.2 kSOAP

kSOAP adalah *library web service* dari klien yang terbatas untuk program-program Java seperti Applets atau aplikasi J2ME (CLDC/CDC/MIDP). kSOAP yang digunakan pada aplikasi ini adalah kSOAP 2 yang merupakan rancangan ulang yang lengkap, dengan mengambil catatan dari kinerja kSOAP 1.x. Beberapa perbedaan antara kSOAP 1.x dengan kSOAP 2 adalah :

1. Struktur yang lebih bebas.
2. kSOAP 2 telah mendukung untuk *literal encoding*.
3. *Support* SOAP Serialization tidak lagi menjadi kebutuhan penting (*optional*) dan terpisah pada *package* yang lain.
4. Beberapa *class-class* yang berbeda telah diintegrasikan ke class SoapSerializationEnvelope, menyediakan dukungan SOAP Serialization. SoapSerializationEnvelope memperluas class dasar SoapEnvelope.

3.3.5 Tools Pengembangan (Mozilla Firefox)

Pemanfaatan *add-ons* Mozilla digunakan dalam pengembangan aplikasi ini. *Add-ons* ini berguna dalam proses development untuk pengujian atau *testing* sebuah *web service* yang kita buat dapat berjalan dengan lancar, sehingga meminimalisir kesalahan-kesalahan dalam proses pengembangan aplikasi ini. Berikut *add-ons* yang diperlukan :

3.3.5.1 JSONView

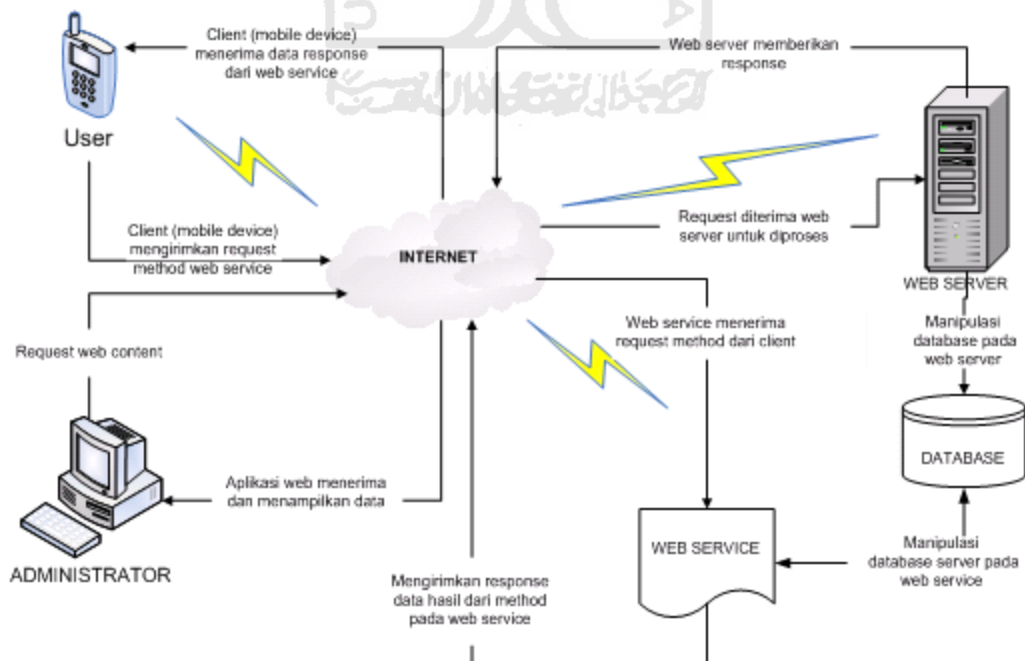
Secara sederhana *add-ons* ini berguna untuk menampilkan file berformat JSON agar menjadi *human readable* atau lebih mudah dalam pembacaan dokumen JSON.

3.3.5.2 SOA Client

Memberikan tampilan yang memudahkan dalam membaca *method-method* yang disediakan oleh SOAP daripada format XML, serta dapat melakukan pengujian pada proses *request* dan memberikan hasil *response*.

3.4 Perancangan Aplikasi *Mobile commerce*

Aplikasi *mobile commerce* menggunakan perangkat *mobile*, PC, *web service*, dan *web server*. Pada sistem ini pelanggan atau member mengakses aplikasi *mobile commerce* melalui perangkat *mobile*. Untuk mengakses *mobile commerce* pada aplikasi web (*server*) diperlukan request terhadap *web service*, kemudian *web service* akan memberikan *response* data yang akan diakses untuk pelanggan. Sedangkan administrator untuk mengaksesnya menggunakan PC melalui aplikasi web. Gambaran arsitektur sistem terlihat pada gambar 3.1



Gambar 3.1 Arsitektur sistem

Berdasarkan arsitektur sistem diatas diperlukan perancangan sistem untuk membangun aplikasi *mobile commerce*, yakni perancangan sistem *web service*, perancangan sistem aplikasi, perancangan masukan sistem, perancangan *database*, dan perancangan tampilan.

3.4.1 Perancangan Sistem *Web service*

3.4.1.1 Sistem *Web service*

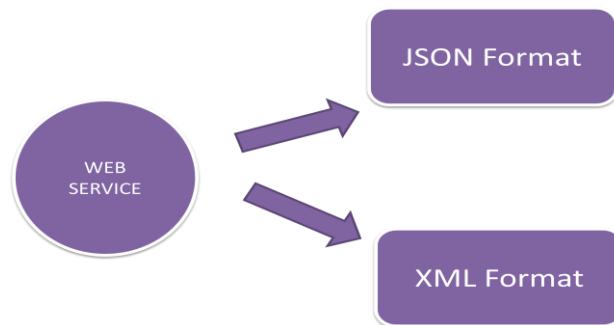
Sistem *web service*, dipergunakan untuk melakukan pertukaran data antara *client* dengan *server*. *Web service* dijadikan jembatan penghubung antara aplikasi (*client*) dengan database (*server*). Secara umum aplikasi *client* melakukan *request* dengan memanggil *method* atau fungsi pada *web service*, yang kemudian *method* tersebut melakukan *query* pada database *server*. Hasil *query* akan memberikan *response* ke *client* melalui *web service*. Gambaran alur *web service* secara umum seperti pada Gambar 3.3 :



Gambar 3.2 Bagan Alur *Web service*

3.4.1.2 Format *Web service*

Pada aplikasi *middleware* memiliki dua format yang dijadikan sebagai *web service*, yaitu JSON dan SOAP (gambar 3.4).



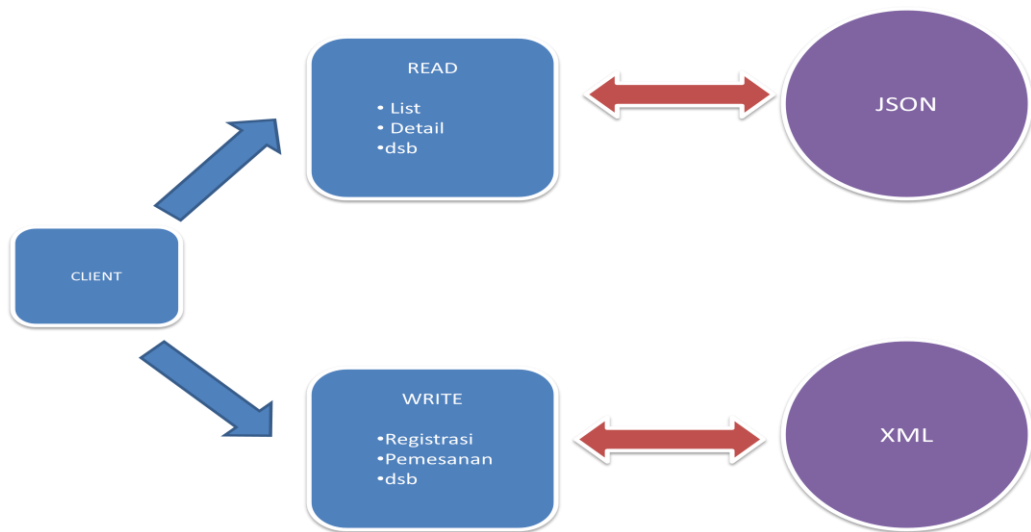
Gambar 3.3 Format *Web service*

Penggunaan kedua komponen *web service* ini didasarkan pada aktifitas *client* dalam melakukan pertukaran data sebagai berikut :

1. **Format JSON** : Aktifitas *read process*, dimana user tidak memerlukan interaksi dalam mengirimkan data ke *server*, seperti daftar dan detail produk, serta melihat data member.
2. **Format SOAP** : Aktifitas *write process*, dimana user memerlukan interaksi untuk mengirimkan data ke *server*, seperti registrasi dan proses pemesanan produk.

3.4.1.3 Aktifitas *Web service*

Proses alur sistem *web service* pada aktifitas *client* merupakan integritas hubungan antara aplikasi *server*, aplikasi *client-server* dan aplikasi *client* (gambar 3.5). Secara detail proses aktifitasnya digambarkan sebagai berikut:



Gambar 3.4 Aktifitas *Web service*

Aktifitas sistem *web service* dapat dijelaskan sebagai berikut :

1. Aplikasi *client* memiliki aktifitas *read* dan *write* untuk melakukan *request*
 - i. Aktifitas *read* adalah aktifitas user yang tidak memerlukan interaksi dalam pengiriman data dari *client* ke *server*. Aktifitasnya adalah :
 - a. Aktifitas *user* melihat daftar produk.
 - b. Aktifitas *user* melihat detail produk.
 - c. Aktifitas *user* melakukan registrasi.
 - d. Aktifitas *user* melihat data member.
 - ii. Aktifitas *write* adalah aktifitas user yang memerlukan interaksi dalam pengiriman data dari *client* ke *server*. Aktifitasnya adalah :
 - a. Aktifitas *user* melakukan pemesanan produk sebagai pelanggan baru.
 - b. Aktifitas *user* melakukan pemesanan produk sebagai member.
 - c. Aktifitas *user* melakukan konfirmasi pesanan.
 - d. Aktifitas *user* mengubah data member.
2. Aplikasi *client-server* mengirimkan *request* ke *server* dengan memanggil *method* atau fungsi *web service* sesuai dengan aktifitas user.
 - i. JSON digunakan untuk melakukan aktifitas *read*, memiliki *method* diantaranya :

- a. JSON *Products* (*request* daftar produk).
 - b. JSON *Product* (*request* detail produk).
 - c. JSON *Orders* (*request* data pesanan member).
 - d. JSON *User* (*request* data member).
- ii. SOAP digunakan untuk melakukan aktifitas *write*, memiliki *method* diantaranya:
- a. *Method setNewOrder* (Fungsi tambah pemesanan untuk pelanggan baru)
 - b. *Method setMemberOrder* (Fungsi tambah pemesanan untuk member)
 - c. *Method setDetailOrder* (Fungsi tambah detail pemesanan)
 - d. *Method setConfirmation* (Fungsi konfirmasi pembayaran pesanan)
 - e. *Method setUserInfo* (Fungsi ubah informasi member)

3.4.2 Perancangan Sistem Aplikasi

Perancangan sistem pada aplikasi *mobile commerce* ini adalah dengan menggunakan UML (*Unified Modelling Language*). UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak berorientasi objek.

Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk diagram/symbol untuk mempresentasikan elemen-elemen dalam sistem. Bentuk diagram yang digunakan untuk mempresentasikan sistem *mobile commerce* sebagai berikut :

3.4.2.1 Use Case Diagram

Use case diagram adalah suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem. Sebuah Use case diagram mempresentasikan interaksi yang terjadi antara aktor dengan proses atau sistem yang dibuat. Use case diagram bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah sistem yang digunakan.

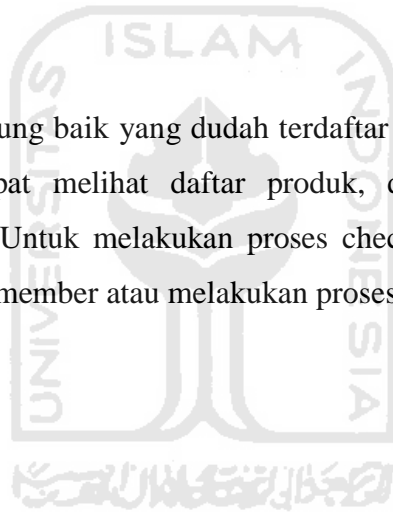
Berdasarkan gambar *use case diagram* (lihat gambar 3.6), dijelaskan bahwa terdapat aktor yang berperan dalam aplikasi ini. Aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu, yaitu:

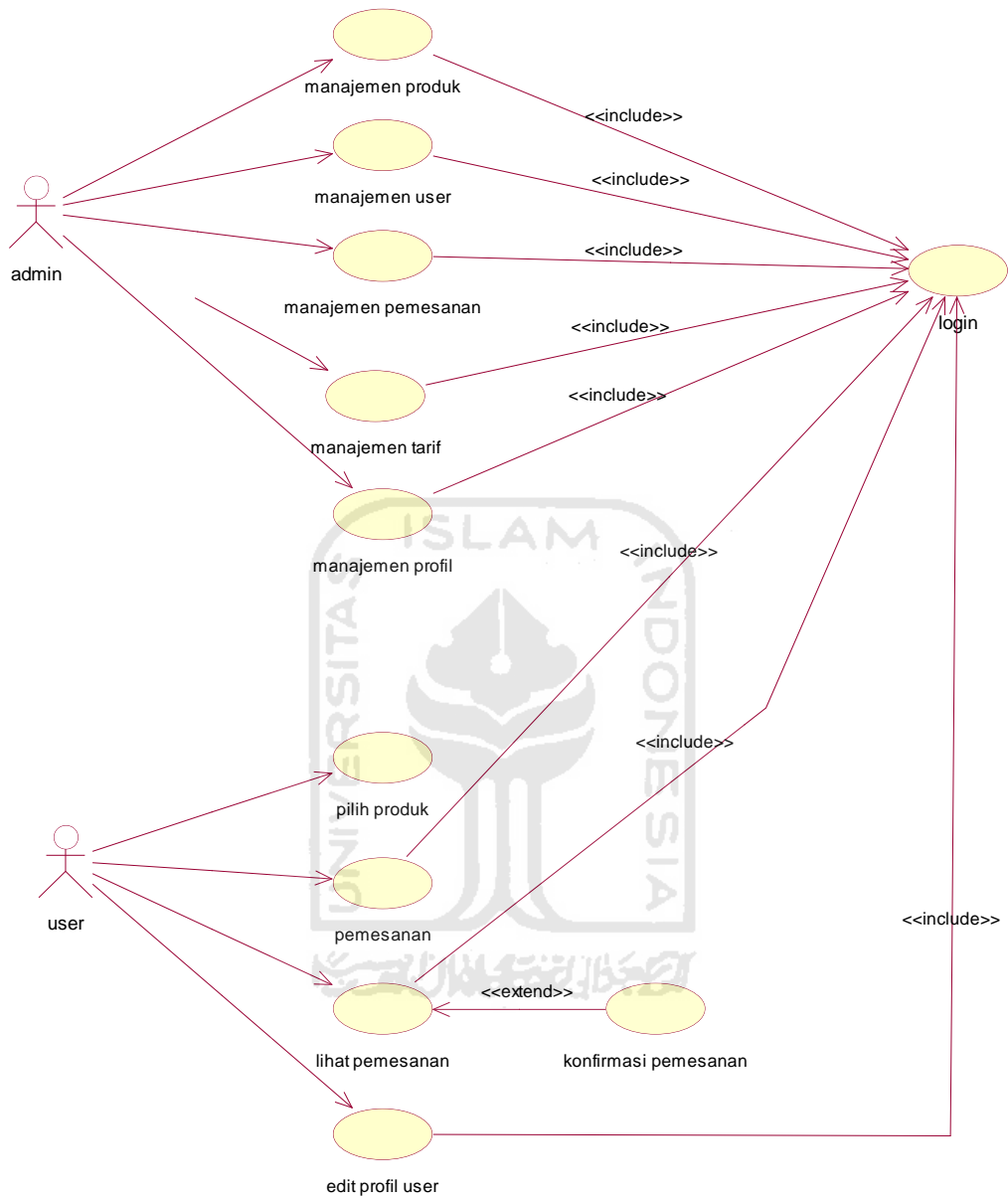
1. Admin

Aktor admin dapat melakukan proses manajemen/pengolahan data produk, pemesanan, user pada sisi *server* melalui sebuah aplikasi web. Diperlukan login sebagai hak akses admin, digambarkan dengan relasi *include* pada *use case* admin.

2. User

Aktor *user*/pengunjung baik yang sudah terdaftar sebagai member ataupun tidak terdaftar dapat melihat daftar produk, detail produk dan lihat keranjang belanja. Untuk melakukan proses checkout pemesanan produk diperlukan sebagai member atau melakukan proses registrasi.





Gambar 3.5 Use Case Diagram

3.4.2.2 Class Diagram

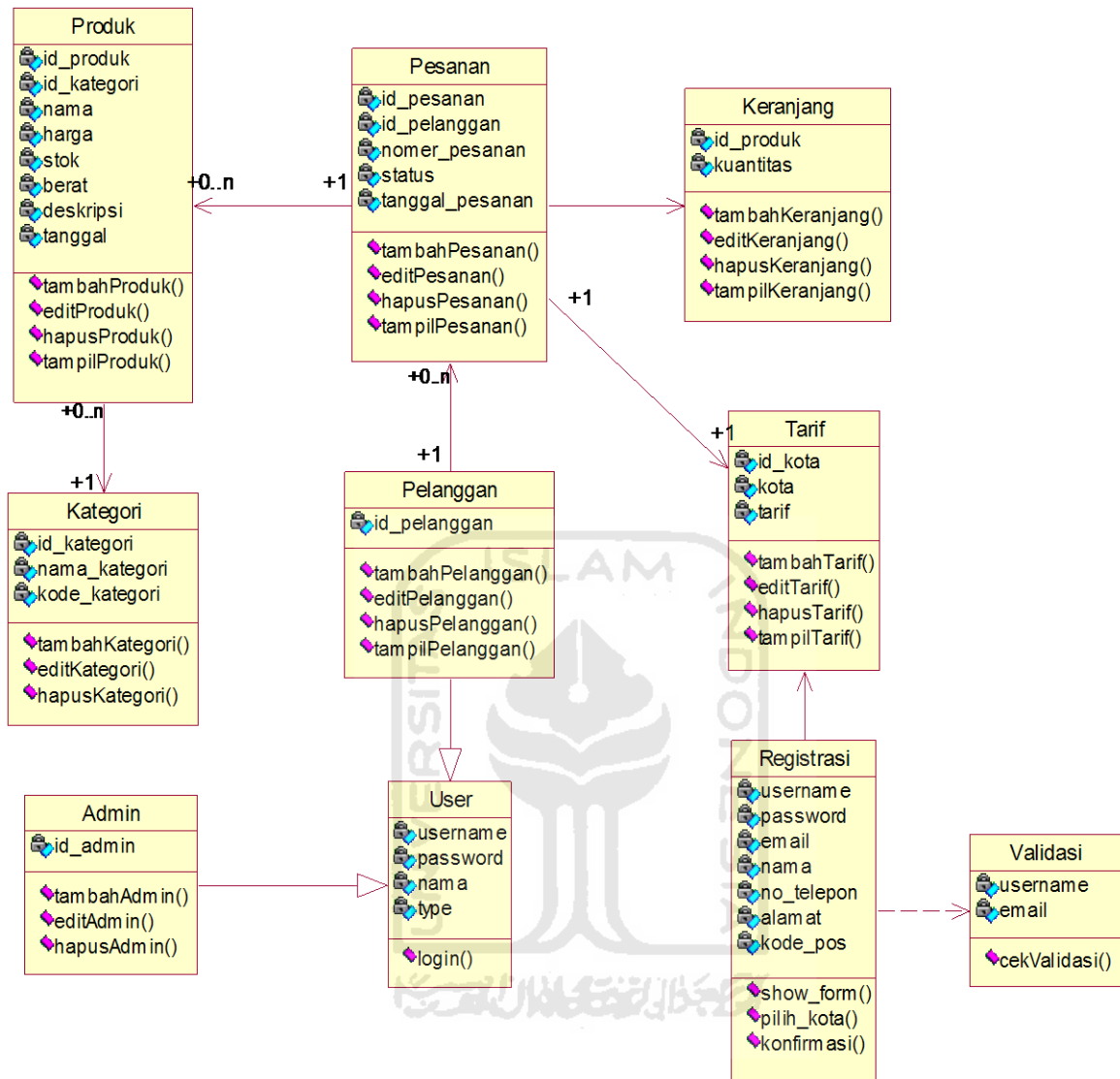
Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan antar-class di dalam sistem seperti containment, pewarisan,osiasi dan lain-lain. Class adalah sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Class adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, behavior umum, relasi umum, dan semantic/kata yang umum. Class-class ditentukan/ditemukan dengan cara memeriksa objek-objek dalam sequence diagram dan collaboration diagram. Sebuah class digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. Class sebaiknya diberi nama menggunakan kata benda sesuai dengan domain/bagian/kelompoknya.

Elemen-elemen class diagram dalam pemodelan UML terdiri dari: Class-class, struktur class, sifat class (class behavior), perkumpulan/gabungan (association), pengumpulan/kesatuan (agregation), ketergantungan (dependency), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan role name (peranan/tugas nama).

Berdasarkan *Class Diagram* (lihat gambar 3.7), dijelaskan sebagai berikut

1. Pesanan berasal dari satu pembeli/pelanggan dan pembeli dapat melakukan beberapa kali pesanan.
2. Class pelanggan/pembeli dan class pesanan adalah relasi **asosiasi**.
3. Relasi antara class pesanan dan class keranjang adalah **asosiasi**.
4. *Class* registrasi memiliki relasi **dependency** terhadap *class* validasi, dimana terdapat proses validasi data yang dilakukan sistem.
5. Terdapat relasi **generalization dan inheritance** yaitu hubungan antara class yang bersifat umum dengan class yang bersifat spesifik, pada *class* pelanggan dan admin terhadap *class* user



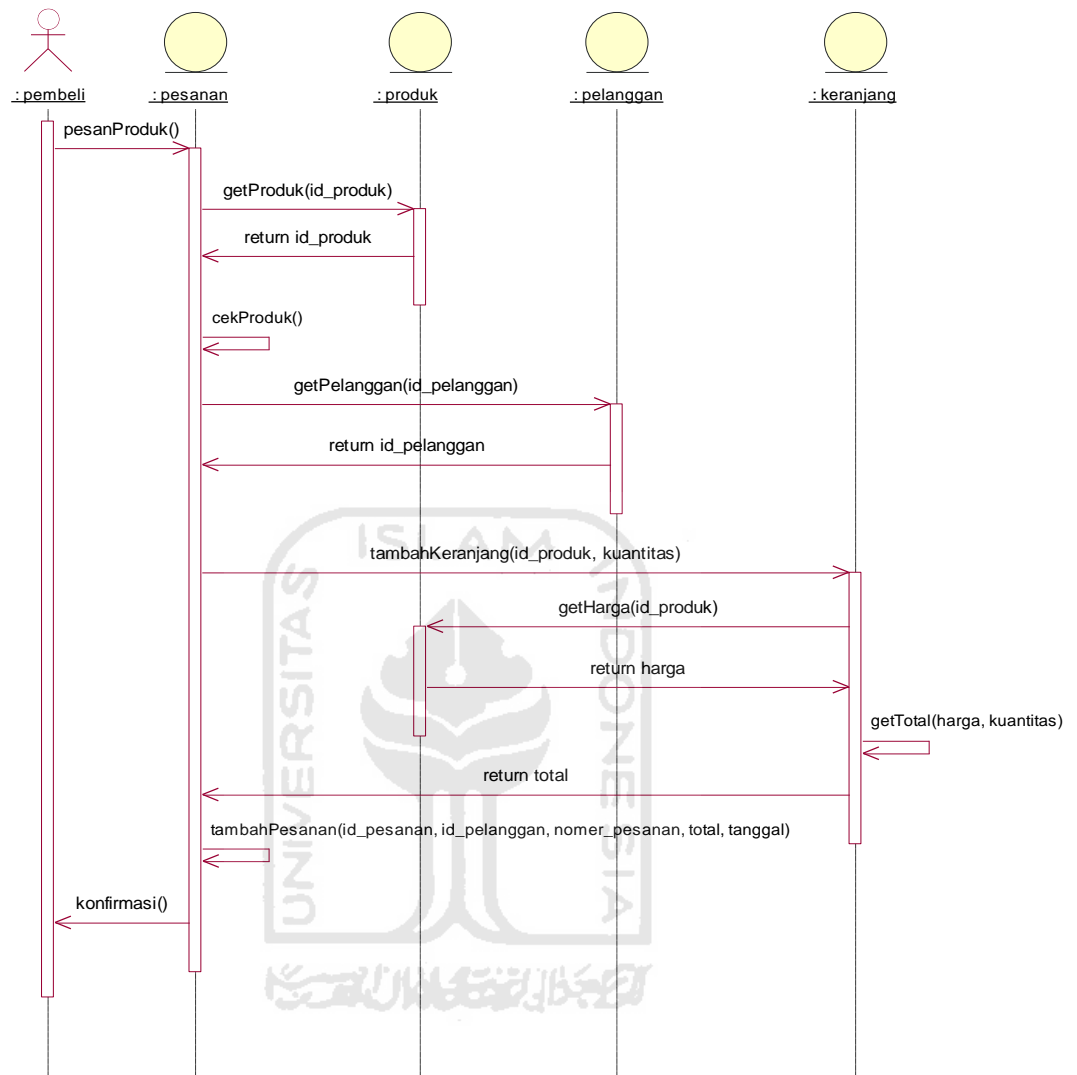
Gambar 3.6 Class Diagram

3.4.2.3 Sequence Diagram

Sequence diagram merupakan gambaran interaksi antar objek dalam dan disekitar sistem yang saling berkomunikasi menggunakan pesan dan memiliki parameter waktu.

Fungsi utama sequence diagram yaitu untuk menggambarkan urutan langkah-langkah yang dilakukan sebagai respon yang memicu aktifitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Pada sequence diagram (lihat gambar 3.8) terlihat bahwa objek pesanan memiliki *method* tambahPesanan untuk melakukan proses pemesanan produk. *Method* tersebut membutuhkan *id_pesanan*, *id_pelanggan*, *nomer_pesanan*, *total* dan *tanggal* yang dapat diperoleh dari object produk, pelanggan, dan keranjang. Sebelumnya pada objek produk memiliki *method* cekProduk, untuk melakukan pengecekan produk yang akan dipesan. Apabila data produk tersedia maka data akan dikirim ke objek keranjang yang memiliki *method* getTotal untuk mendapatkan data total guna proses pemesanan produk. Selanjutnya dari objek pesanan akan mengirimkan konfirmasi kepada pembeli.

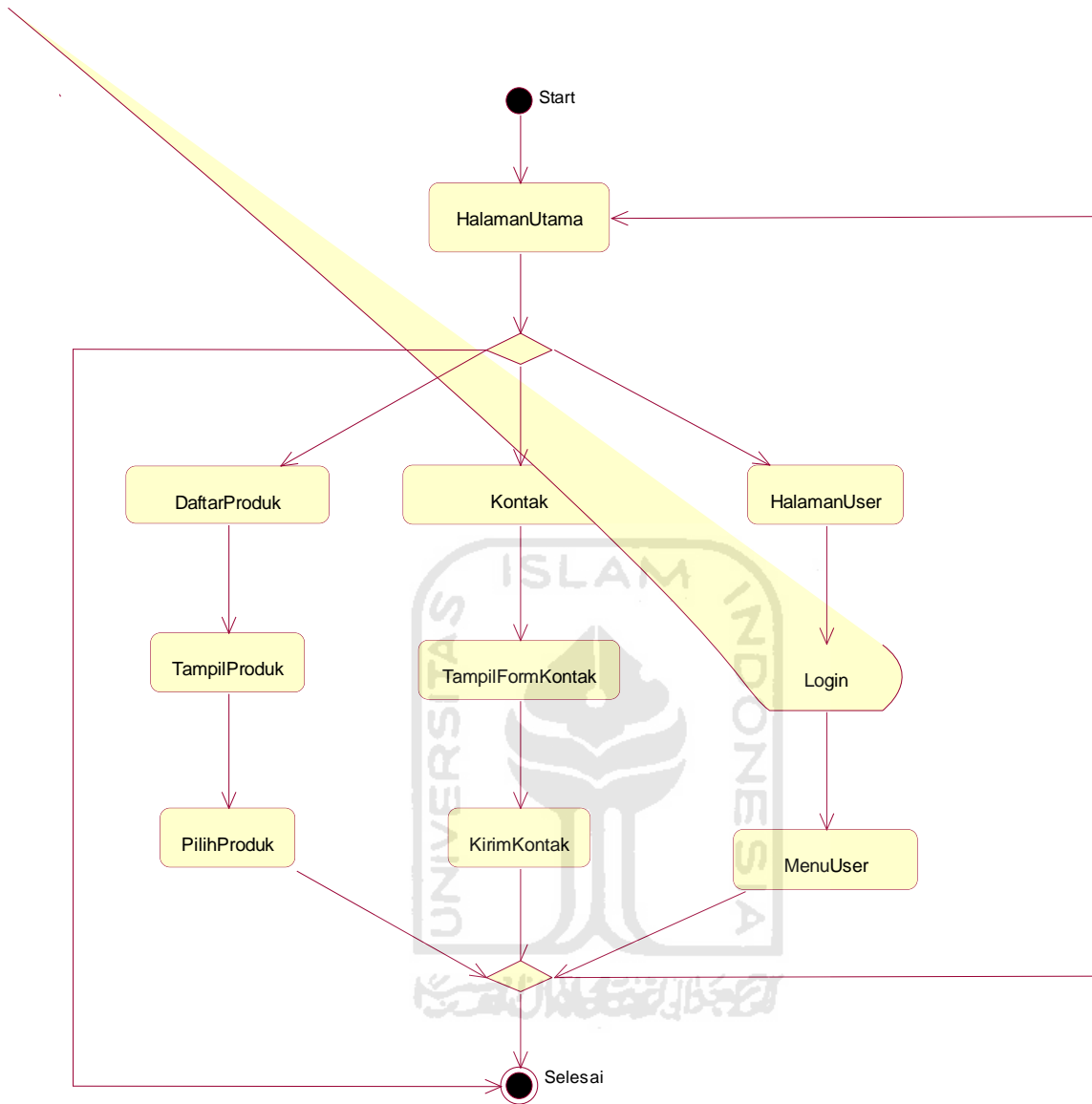


Gambar 3.7 Sequence Diagram

3.4.2.4 Activity Diagram

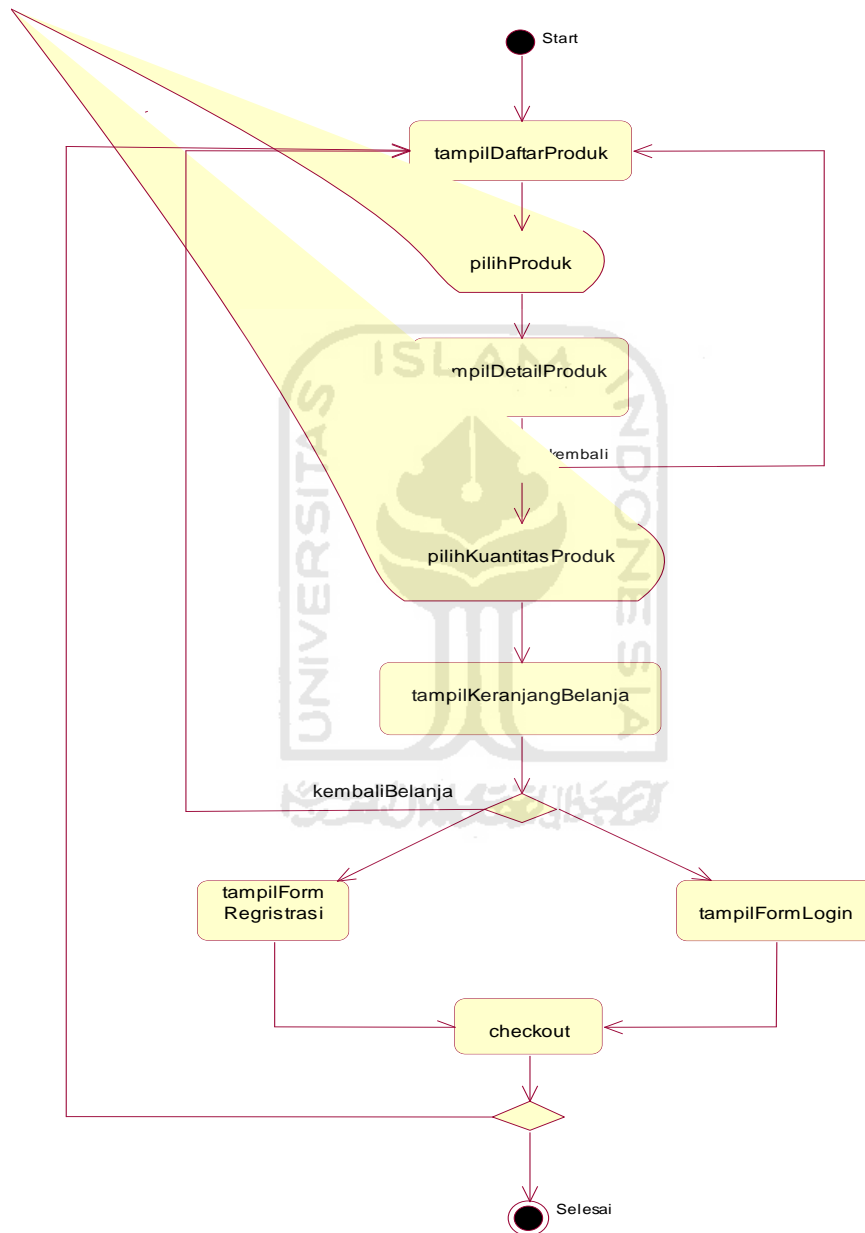
Activity diagram adalah representasi grafis dari alur kerja tahapan aktifitas. Diagram ini mendukung pilihan tindakan, iterasi dan *concurrency*. Pada pemodelan UML, *activity diagram* dapat digunakan untuk menjelaskan bisnis dan alur kerja operasional secara *step-by-step* dari komponen suatu sistem. *Activity diagram* menunjukkan keseluruhan dari aliran kontrol.

Activity Diagram User merupakan alur aktifitas user dalam sistem. Gambar dibawah menjelaskan activity diagram user.



Gambar 3.8.1 Activity Diagram User

Activity Diagram Pembelian merupakan alur aktifitas pembeli/pelanggan melakukan pemesanan produk pada sistem, seperti pada gambar dibawah ini :



Gambar 3.8.2 Activity Diagram Pembelian User

3.4.3 Perancangan Masukan Sistem

Perancangan masukan sistem pada aplikasi *mobile commerce* ini adalah berupa data yang berkaitan dengan aktifitas sistem seperti proses registrasi, login, pemesanan, konfirmasi pesanan dan perubahan data member.

3.4.3.1 Registrasi

Proses registrasi dimana user melakukan pendaftaran untuk menjadi member guna melakukan proses pemesanan produk pada sistem. Pada proses ini data masukan berupa username, password, nama, email, telepon, alamat, kota, dan kodepos.

3.4.3.2 Login

Proses login diperlukan user untuk menjadikan hak akses sebagai member. Selain berguna untuk melakukan pemesanan produk, hak akses member dapat melihat *history* pesanan dan konfirmasi pesanan, serta juga dapat melakukan perubahan data. Pada proses login data masukan berupa username dan password.

3.4.3.3 Pemesanan produk

Proses pemesanan produk yang dilakukan user sebagai member dengan data masukan berupa data pelanggan dan data produk, serta biaya pesanan.

3.4.3.4 Konfirmasi Pesanan

Proses konfirmasi hanya dilakukan user yang memiliki hak akses sebagai member. Pada proses konfirmasi memiliki 2 tipe, yaitu konfirmasi pembayaran dan konfirmasi penerimaan pesanan. Pada proses ini data masukan berupa data pesanan user (id_pesanan).

3.4.3.5 Perubahan Data Member

Proses perubahan data member hanya dilakukan user yang memiliki hak akses sebagai member. Member dapat merubah data profil seperti merubah nama, email, atau alamat. Pada proses ini data masukan berupa data profil member.

3.4.4 Perancangan Database

Database bertugas untuk memberikan informasi serta penyimpanan data bagi user. Dalam membangun aplikasi *mobile commerce* ini, diperlukan pengambilan data dari *server* serta pengiriman data dari *client* ke *server*. Selain penggunaan *database* dari *server* juga diperlukan *database client* diperuntukan menyimpan data dari *server* pada *client* melalui *web service*

3.4.4.1 MySQL

Pada aplikasi *server*, database yang digunakan dengan menggunakan MySQL. Database pada *server* ini digunakan dalam pembuatan *web service*, berikut adalah tabel-tabel yang digunakan :

Tabel *Categories*, untuk membagikan produk berdasarkan kategori

Tabel 3.1 Tabel categories

Field	Tipe Data
category_id	bigint(20)
category_name	varchar(255)
code	varchar(5)

Tabel *Products*, untuk menyimpan data produk

Tabel 3.2 Tabel products

Field	Tipe Data
product_id	bigint(20)
category_id	bigint(20)
title	varchar(255)
url	varchar(255)
price	mediumint(8)
stock	tinyint(3)
weight	tinyint(3)

description	text
vote	mediumint(9)
rate	mediumint(9)
ts_created	datetime
status	char(1)

Tabel *Products Images*, untuk menyimpan data gambar dari produk.

Tabel 3.3 Tabel products_images

Field	Tipe Data
image_id	bigint(20)
filename	varchar(255)
product_id	bigint(20)
ranking	int(10)

Tabel *Products Comments*, untuk menyimpan data komentar pada sebuah produk

Tabel 3.4 Tabel products_comments

Field	Tipe Data
comment_id	bigint(20)
product_id	bigint(20)
comment_author	varchar(255)
comment_author_email	varchar(255)
comment_author_url	varchar(255)
comment_author_IP	varchar(255)
comment_message	text
comment_created	datetime
comment_approved	tinyint(4)
comment_parent	bigint(20)
user_id	bigint(20)

Tabel *Users*, untuk menyimpan data akun pelanggan, dalam melakukan autentikasi login, serta tipe pelanggan.

Tabel 3.5 Tabel users

Field	Tipe Data
user_id	bigint(20)
username	varchar(255)
password	varchar(32)
user_type	varchar(20)
ts_created	datetime
ts_last_login	datetime

Tabel *Users Profile*, untuk menyimpan data diri pribadi pelanggan, serta untuk menyimpan *token* yang berguna untuk melakukan otentikasi pada sistem *mobile commerce*.

Tabel 3.6 Tabel users_profile

Field	Tipe Data
user_id	bigint(20)
profile_key	varchar(255)
profile_value	text

Tabel *Users Shipping*, untuk menyimpan data pengiriman atau alamat pelanggan untuk mengirimkan pesanan.

Tabel 3.7 Tabel users_shipping

Field	Tipe Data
user_id	bigint(20)
profile_key	varchar(255)
profile_value	text

Tabel *Orders*, untuk menyimpan data pesanan pelanggan.

Tabel 3.8 Tabel orders

Field	Tipe Data
order_id	bigint(20)
user_id	bigint(20)
code	char(7)
amount	mediumint(9)
status	tinyint(3)
ts_created	datetime

Tabel *Orders Profile.*, untuk menyimpan data profil status pesanan pelanggan, termasuk konfirmasi pesanan.

Tabel 3.9 Tabel orders_profile

Field	Tipe Data
order_id	bigint(20)
profile_key	varchar(255)
profile_value	text

Tabel *Orders Detail*, untuk menyimpan data detail pesanan produk yang dipesanan oleh pelanggan.

Tabel 3.10 Tabel orders_detail

Field	Tipe Data
order_id	bigint(20)
product_id	bigint(20)
qty	tinyint(3)

Tabel *Options*, untuk menyimpan data profil dan *setting* perusahaan.

Tabel 3.11 Tabel options

Field	Tipe Data
option_id	bigint(20)
profile_key	varchar(255)
profile_value	text

Tabel *Options Banks.*, untuk menyimpan data bank-bank perusahaan yang berguna untuk proses transaksi pembayaran melalui transfer

Tabel 3.12 Tabel options_banks

Field	Tipe Data
bank_id	bigint(20)
bank	varchar(255)
rekening_name	varchar(255)
rekening_nomor	varchar(255)

Tabel *Cities*, untuk menyimpan data biaya pengiriman berdasarkan tempat kota, serta berguna dalam proses registrasi.

Tabel 3.13 Tabel cities

Field	Tipe Data
city_id	bigint(20)
province_id	bigint(20)
city	varchar(50)
tax	mediumint(9)
status	tinyint(3)

Tabel *Provinces*, untuk menyimpan data biaya pengiriman berdasarkan tempat provinsi, serta berguna dalam proses registrasi.

Tabel 3.14 Tabel provinces

Field	Tipe Data
province_id	bigint(20)
region	varchar(50)
province	varchar(50)
status	tinyint(3)

3.4.4.2 BlackBerry Persistent Store

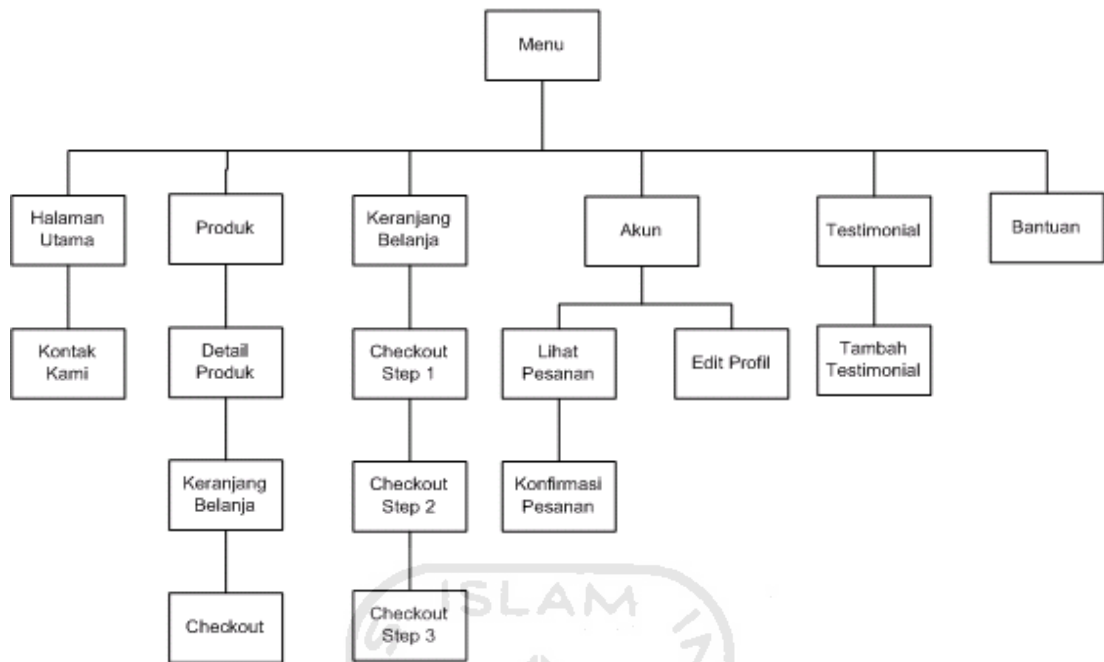
Pada aplikasi *client*, penggunaan *database* ditujukan untuk menyimpan data yang dikirim oleh *web service* dari *server*. Penyimpanan data pada BlackBerry sebagai objek ke memori flash menggunakan BlackBerry Persistent Store. Dengan BlackBerry Persistent Store API, dapat menyimpan seluruh objek Java ke memori tanpa harus *serialize* data terlebih dahulu yang dirancang untuk menyediakan antarmuka penyimpanan data yang fleksibel dan kuat.

3.4.5 Perancangan Tampilan

Perancangan tampilan berguna untuk menghasilkan tampilan layar atau *interface* aplikasi yang interaktif dihasilkan oleh aplikasi *mobile commerce*. Adapun 2 (dua) perancangan tampilan yaitu rancangan antar muka *user* dan rancangan antar muka admin.

3.4.5.1 Rancangan Antar Muka User

Rancangan antar muka *user* merupakan halaman bagi *user* untuk berinteraksi terhadap sistem ini. Adapun rancangan tampilan *client user* dapat dilihat pada Gambar 3.9



Gambar 3.9 Rancangan antar muka user

Keterangan rancangan tampilan *client user* sebagai berikut :

1. Halaman Utama

Pada halaman utama akan ditampilkan informasi tentang visi misi, alamat perusahaan dan juga terdapat menu kontak yang berguna bagi pelanggan untuk memberikan kritik dan saran melalui email.

2. Menu Produk

Pada menu produk, ditampilkan tentang daftar produk yang akan dipilih oleh *user*. Menampilkan detail produk yang dipilih oleh *user* didalamnya berisikan harga, deskripsi, jumlah stok yang tersedia, deskripsi produk serta komentar produk (jika ada). Setelah *user* memilih jumlah/kuantitas produk yang dipesan, akan disimpan sementara dalam keranjang belanja.

3. Menu Keranjang Belanja

Pada menu keranjang belanja memberikan tampilan daftar produk yang dipesan *user*, serta dapat merubah kuantitas/jumlah ataupun menghapus produk yang dipesan.

4. Menu Akun

Pada menu akun diperlukan hak member dengan login terlebih dahulu. Sebagai member dapat melihat status pesanan, serta melakukan konfirmasi pembayaran dan logout.

5. Menu Testimonial

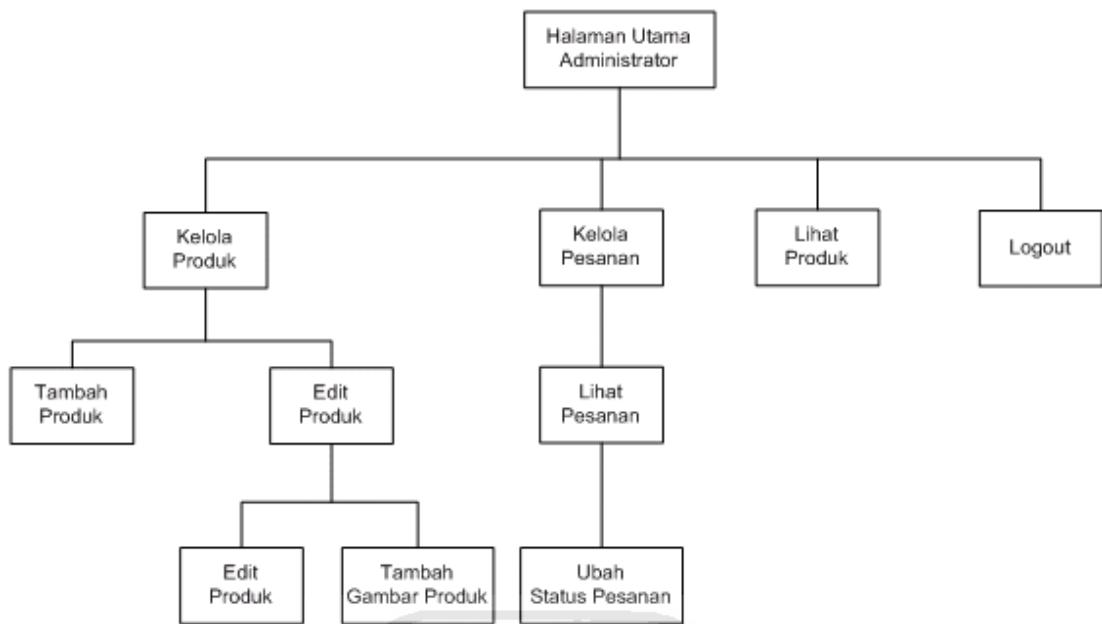
Pada menu testimonial *user* dapat melihat dan memberikan pesan/kesan mereka selama berbelanja di barbequa.

6. Menu Bantuan

Menu bantuan ditunjukkan untuk memberikan bantuan terhadap *user*, diantaranya tentang tata cara berbelanja, cara pembayaran, dan konfirmasi pembayaran.

3.4.5.2 Rancangan antar Muka Admin

Rancangan antar muka admin ini adalah tampilan halaman admin pada aplikasi *mobile commerce*. Halaman antar muka admin pada *mobile commerce* memiliki fungsionalitas lebih sederhana dari pada halaman admin pada aplikasi web. Dalam halaman tersebut, admin hanya dapat melakukan aktifitas untuk mengelola data produk dan data pemesanan.



Gambar 3.10 Rancangan antar muka admin

Pada gambar rancangan antar muka admin di atas terdapat tiga menu utama yaitu:

1. Halaman Utama Administrator

Halaman utama admin berisikan tentang informasi terkini yang berhubungan dengan sistem admin dalam kuantitas, yakni data member baru, data pemesanan baru, testimonial yang masuk dan keuntungan pemasukan.

2. Menu Kelola Produk

Pada menu kelola produk admin dapat mengolah data produk seperti menambah produk, edit produk dan menambah gambar produk.

3. Menu Kelola Pesanan

Pada menu kelola pesanan, admin dapat melihat pesanan dan merubah status pesanan.

BAB IV

IMPLEMENTASI

4.1 Implementasi Sistem

Pada tahap implementasi, sistem telah dirancang agar dapat digunakan dalam keadaan yang sebenarnya. Melalui tahap ini dapat dipastikan apakah sistem telah mencapai tujuan awal seperti yang diinginkan. Untuk mengetahui kesalahan-kesalahan pada sistem yang dibangun maka diperlukan pengujian terhadap sistem tersebut.

4.1.1 Batasan Implementasi

1. Sistem aplikasi *mobile commerce* ini akan diakses dengan Blackberry OS 4.6 keatas.
2. Sistem pembayaran dilakukan secara *offline* atau *transfer*
3. Sistem tidak membahas proses pengiriman

4.1.2 Implementasi *Web service*

Pada aplikasi *mobile commerce* ini diperlukan implementasi sebuah *web service* yang berguna sebagai akses jembatan antara aplikasi *server* dengan *client* dalam pertukaran data. Seperti sudah dibahas di bab sebelumnya, pembuatan *web service* dibagi menjadi dua format, yakni JSON dan SOAP.

4.1.2.1 *Web service* JSON

Pembuatan *web service* JSON didasarkan atas aktivitas *read process user*. Kemampuan JSON bermanfaat untuk efisiensi transfer data, karena hanya membutuhkan sedikit karakter untuk inialisasi variabel array tersebut. Sehingga mudah dalam memarsing /mengolah data yang diberikan oleh JSON.

Dengan memanfaatkan *add-ons* JSONView pada browser Mozilla Firefox, dokumen JSON akan lebih mudah terbaca (human

readable). Sehingga memudahkan untuk mengetahui dan mengolah informasi yang diberikan oleh server melalui JSON.

Implementasi *web service* JSON dilakukan pada aplikasi web, yang secara teknis untuk melakukan *encoding* JSON dengan fungsi `json_encode()` pada PHP. Berikut adalah daftar *web service* JSON yang digunakan :

1. JSON Daftar Produk

Mengirimkan data daftar produk. Berikut tabel dan kolom pada database server yang digunakan untuk membuat JSON Daftar Produk :

Tabel	Kolom
categories	category_name
products	product_id, title, description, price, created
products_images	image_id, filename
products_comments	Count(jumlah komentar)

Dokumen JSON ini akan diakses pada menu tampilan daftar produk pada *mobile commerce*. Gambar JSON daftar produk terlihat pada lampiran 4.1.

2. JSON Detail produk

Menampilkan data detail dan komentar (jika tersedia) pada produk dengan parameter id produk (`id_produk`). Berikut tabel dan kolom pada database server yang digunakan untuk membuat JSON Detail produk :

Tabel	Kolom
categories	category_id, category_name
products	product_id, title, price, stock, size, rate, description, created
products_images	image_id, filename
products_comments	comment_author,comment_author_email,comment_

	message
--	---------

Dokumen JSON ini akan diakses pada menu tampilan detail produk pada *mobile commerce*. Gambar JSON daftar produk terlihat pada lampiran 4.2

3. JSON Tarif

Menampilkandata kota, provinsi beserta tarif. Dokumen JSON ini akan diakses pada menu tampilan registrasi *user* dan menu perubahan data anggota pada *mobile commerce*. Pemilihan kota bertujuan untuk menambahkan biaya pengiriman sesuai dengan jarak dalam lingkup kota, yang akan dihitung oleh sistem terlihat pada lampiran 4.3.

. Berikut tabel dan kolom pada database server yang digunakan untuk membuat JSON Tarif:

Tabel	Kolom
provinces	province
cities	city_id, city, tax

4. JSON Informasiuser

Menampilkan data informasi *user/anggota*. Untuk mengakses dokumen ini, diperlukan sebuah token yang didapatkan dari hasil login. Token ini berisikan kode hash yang dibangkitkan oleh sistem jika *user* telah terotentikasi melalui aksi login. Penggunaan token bertujuan untuk keamanan terhadap data informasi pelanggan, sehingga data tetap aman dan rahasia, juga tidak semua orang bisa mengakses dokumen ini terkecuali *user* yang telah terotentikasi. Berikut tabel dan kolom pada database server yang digunakan untuk membuat JSON Informasiuser:

Tabel	Kolom
users_profile	access_token,first_name, last_name, email, phone
users_shipping	address, address2, postal
cities	city_id, city, tax

Dokumen JSON ini akan diakses pada menu tampilan *checkout* sebagai anggota, dan menu akun pada *mobile commerce*. Gambar JSON informasi *user/anggota* yang telah terotentikasi terlihat pada lampiran4.4.1, dan apabila *user* biasa (tidak ter-otentikasi) maka akan menampilkan kesalahan seperti pada lampiran 4.4.2

5. JSON Pemesanan *User*

Menampilkan data pemesanan pelanggan beserta detail pemesanan dan status pesanan. Sama halnya dengan dokumen JSON informasi user, untuk mengakses dokumen ini diperlukan sebuah token yang didapatkan dari hasil login. Berikut tabel dan kolom pada *database server* yang digunakan untuk membuat JSON Pemesanan *user*:

Tabel	Kolom
users_profile	access_token
orders	order_id, code, amount, status, created
orders_detail	qty
products	title

Dokumen JSON ini akan diakses pada menu tampilan menu lihat pesanan dan konfirmasi pesanan pada *mobile commerce*. Gambar JSON informasi user yang telah terotentikasi terlihat pada lampiran 4.5.1, dan apabila user biasa (tidak ter-otentikasi) maka akan menampilkan kesalahan seperti pada lampiran4.5.2

4.1.2.2 Web service SOAP

Implementasi *web service* SOAP dikembangkan menggunakan *library Zend Framework* dan *library SOAP Extension* pada *PHP.Library* ini menyediakan kelas-kelas yang dapat digunakan untuk membuat *web service*. Kelas *Zend_Soap_Server* digunakan untuk membuat *instance server web service* yang berfungsi untuk menyediakan *web service (service provider)* atau sebagai *listener* yang akan menerima dan merespon permintaan akses terhadap *web service*.

Aplikasi *mobile commerce* mendapatkan deskripsi layanan yang dimiliki *server (web)* melalui dokumen WSDL. Dengan WSDL ini, aplikasi *mobile commerce* dapat mengetahui deskripsi layanan yang dimiliki berupa *method* apa saja yang tersedia, parameter masukan, dan parameter keluaran dari *method* tersebut. Dokumen WSDL ini tidak dapat dibangkitkan secara otomatis oleh *SOAP Extension*. Oleh karena itu, dokumen WSDL akan dibuatkan oleh kelas *Zend_Soap_AutoDiscovery* yang disediakan oleh *library Zend Framework*. Dan *service-service* dari *web service* SOAP terlihat pada tabel 4.1

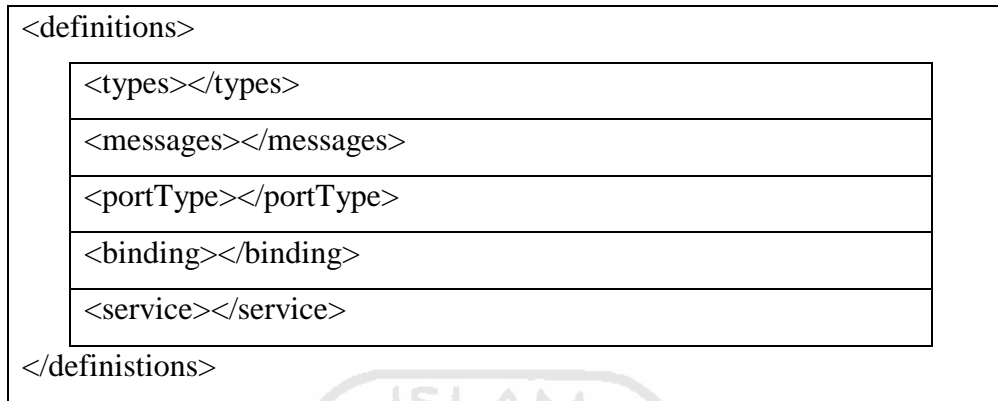
No	Nama Service	Deskripsi	Input	Output
1	setNewOrder	<i>Service</i> ini berfungsi untuk mengirimkan pesanan baru, artinya <i>user</i> melakukan pemesanan dengan status belum pernah berbelanja atau belum terdaftar (<i>member</i>) sebelumnya. Output	username password first_name last_name email phone address address2 city_id postal amount	<i>Array</i> Id Code Total
2	setMemberOrder	<i>Service</i> ini berfungsi	user_id	<i>Array</i>

		untuk mengirimkan pesanan untuk <i>member</i> , artinya <i>user</i> melakukan pemesanan dengan status pernah berbelanja atau sudah terdaftar (<i>member</i>)	amount	Id Code Total
3	setDetailOrder	<i>Service</i> ini berfungsi untuk mengirimkan detail pemesanan, yaitu produk dan kuantitas yang dipesan	order_id product_id qty	Tidak ada
4	setUserInfo	<i>Service</i> ini berfungsi untuk ubah detail informasi <i>member</i> . Diperlukan hak akses member dengan mengirimkan sebuah <i>token</i> yang didapat dari hasil otentikasi (login)	token user_id first_name last_name email phone address address2 city postal	<i>String</i> Aksi informasi
5	setConfirmation Payment	<i>Service</i> ini berfungsi untuk konfirmasi pembayaran pesanan <i>member</i> . Diperlukan hak akses member	token user_id order_id bank rekening	<i>String</i> Aksi informasi

		dengan mengirimkan sebuah <i>token</i> yang didapat dari hasil otentikasi (login)	amount	
6	setConfirmation Delivery	<i>Service</i> ini berfungsi untuk konfirmasi pengiriman pesanan <i>membertelah</i> sampai. Diperlukan hak akses member dengan mengirimkan sebuah <i>token</i> yang didapat dari hasil otentikasi (login)	token user_id order_id	<i>String</i> Aksi informasi
7	getValidation	<i>Service</i> ini berfungsi untuk melakukan validasi ketersediaan username dan email saat proses registrasi pemesanan baru	username email	<i>String</i> Aksi informasi
8	setTestimonial	<i>Service</i> ini berfungsi untuk mengirimkan pesan testimoni baik untuk tamu atau <i>member</i>	author username message	<i>String</i> Aksi informasi

Tabel 4.1 Service Soap

Sebuah dokumen WSDL yang menjelaskan informasi detail sebuah *web service* dalam format XML yang ditandai dengan tag-tag tertentu (Gambar 4.6.1). Dan untuk keterangan setiap tag-tag WSDL dapat dilihat pada tabel 4.1



Gambar 4.6.1 Skema Dokumen WSDL

TAG	KETERANGAN
definitions	Tag tertinggi dalam dokumen WSDL atau bisa dibilang tag root. Tag-tag lainnya harus berada di dalam tag ini. Salah satu yang ada di dalam tag ini adalah <i>namespace</i>
types	Mendefinisikan tipe-tipe data yang digunakan oleh web service. Akan tetapi yang didefinisikan hanyalah tipe-tipe data kompleks seperti <i>array</i> atau <i>struct</i> . Sedangkan tipe-tipe data sederhanayang sudah built-in seperti string, integer, double, dll tidak didefinisikan lagi
messages	Parameter <i>input</i> atau <i>output</i> apa saja yang digunakan
portType	Mendefinisikan <i>method</i> atau fungsi yang disediakan web service
binding	Informasi dalam tag ini adalah bagaimana sebuah operasi atau fungsi diimplementasikan. Apakah fungsi tersebut dapat dipanggil menggunakan protokol HTTP atau SMTP. Informasi seperti soap action juga didefinisikan didalam tag ini
service	Mendefisikan lokasi <i>web service</i>

Tabel 4.2 Keterangan tag WSDL

Mengakses *web service* SOAP diperlukan untuk mengetahui *method-method* yang dikirimkan oleh SOAP melalui WSDL terlihat pada lampiran gambar 4.6.2. Disini diasumsikan atau diposisikan sebagai seorang pengembang yang memanfaatkan *web service* yang sudah dibuat pihak ketiga (*third-party*), Walaupun secara teoritis telah membangun *web service* SOAP, yang tentunya mengetahui *method-method* yang disediakan SOAP. Maka dari itu diperlukan kemampuan untuk membaca file WSDL.

Dengan memanfaatkan *add-ons SOAClient*, membantu mengetahui *method* yang tersedia, beserta *parameterinput* dan *output* pada dokumen WSDL. Selain itu juga *add-ons* ini digunakan sebagai percobaan terhadap sebuah *web service* SOAP yang dibangun. Sehingga meminimalisasi kesalahan dalam proses pengembangan sistem (lampiran gambar 4.6.3).

4.2 Pengujian Sistem

Sistem yang telah dibangun perlu diuji terlebih dahulu. Hal ini diperlukan untuk mengumpulkan kesalahan-kesalahan yang mungkin terdapat dalam sistem tersebut untuk selanjutnya dapat diperbaiki.

4.2.1 Lingkungan Pengujian

1. Software Eclipse IDE versi 3.4 (Ganymede) for Java Developers
2. Blackberry JDE Component Package 4.6.0 / Blackberry OS 4.6
3. Pengujian dengan MDS Simulator Blackberry Bold 9000

4.2.2 Pengujian Sistem User

Pada pengujian ini dilakukan pada halaman *user*, yang merupakan halaman sistem yang dapat diakses oleh *user* atau konsumen dalam melakukan proses pembelian.

4.2.2.1 Halaman Menu Utama (Home)

Pada halaman utama berisikan informasi tentang perusahaan atau toko yang bersangkutan meliputi email, kontak, alamat dan deskripsi. Selain itu

juga terdapat menu “kontak kami”, sehingga user dapat memberikan kritik, saran atau pertanyaan kepada perusahaan tersebut melalui email. Gambaran halaman utama seperti pada lampiran gambar 4.7

4.2.2.2 Halaman Daftar Produk

Pada halaman daftar produk menampilkan daftar-daftar produk yang berisikan informasi singkat produk seperti gambar produk, rating produk, harga dan tanggal terbit produk. Dan juga pada halaman ini terdapat keranjang belanja mini yang bertujuan memberikan informasi kepada user terhadap barang yang sudah dipesan. Tampilan halaman daftar produk terlihat pada lampiran gambar 4.8

4.2.2.3 Halaman Detail Produk

Halaman detail produk menampilkan informasi secara detail produk yang dipilih oleh *user*, seperti jumlah stok yang tersedia, berat produk, deskripsi produk serta komentar produk. Pada halaman inilah *user* melakukan pemesanan produk yang mereka pilih dengan memilih jumlah/kuantitas produk tersebut. Setelah *user* melakukan pemesanan, produk yang di pesan akan disimpan sementara pada keranjang belanja. Tampilan halaman detail produk terlihat pada lampiran gambar 4.9

4.2.2.4 Halaman Keranjang Belanja

Halaman keranjang belanja menampilkan daftar pesanan *user* yang berisikan informasi jumlah/kuantitas pesanan dan harga total keseluruhan produk yang dipesan. Pada halaman ini, *user* dapat melakukan perubahan jumlah/kuantitas ataupun menghapus pesanan.

4.2.2.5 Halaman Checkout Akun

Halaman *checkout* merupakan proses pemesanan setelah *user* selesai berbelanja atau memilih-milih produk yang akan dipesan. Halaman *checkout* terbagi menjadi 3 proses yaitu, *checkoutstep 1*, *checkoutstep 2* dan *checkoutstep3*. Pada halaman *checkout* akun (*step 1*) *user* dapat melakukan proses *checkout* sebagai anggota dengan melakukan login,

ataupun *checkout* sebagai pelanggan baru dengan cara melakukan *registrasi*/pendaftaran. Tampilan halaman *checkout* akun terlihat pada lampiran gambar 4.11

4.2.2.6 Halaman *Checkout* Registrasi

Halaman *checkout* registrasi ditujukan kepada *user* yang merupakan pelanggan baru untuk melakukan *checkout* dengan mengisi form pendaftaran. *User* yang telah melakukan pendaftaran secara otomatis akan terdaftar menjadi anggota, sehingga jika *user* tersebut ingin melakukan proses pemesanan kembali maka *user* tersebut dapat melakukan *fast checkout* yaitu dengan melakukan login. Tampilan halaman *checkout* registrasi terlihat pada lampiran gambar 4.12

4.2.2.7 Halaman *Checkout* sebagai Anggota

Halaman *checkout* sebagai anggota ditujukan kepada *user* yang telah terdaftar atau anggota. Setelah melakukan proses otentikasi pada halaman login (lampiran gambar 4.13.1), pada halaman *checkoutstep* 2 akan menampilkan informasi *user*. Halaman *checkout* sebagai anggota terlihat pada lampiran gambar 4.13.2

4.2.2.8 Halaman *Checkout* Selesai

Halaman *checkoutstep* 3 merupakan proses *checkout* terakhir, menampilkan informasi berupa kode pesanan, total harga yang harus dibayarkan, serta petunjuk cara pembayaran dan konfirmasi pembayaran terlihat pada lampiran gambar 4.14.

4.2.2.9 Halaman Akun Anggota

Halaman akun anggota ditujukan hanya untuk *user* yang memiliki hak akses anggota melalui proses otentikasi pada halaman login (gambar 4.15.1). Pada menu akun ini anggota dapat melihat *history* dan status pesanan, selain itu anggota dapat mengubah data informasi. Tampilan halaman akun anggota terlihat pada lampiran gambar 4.15.2

4.2.2.10 Halaman Konfirmasi Pesanan Anggota

Halaman konfirmasi pesanan anggota dibagi menjadi dua bagian, yakni konfirmasi pembayaran dan konfirmasi penerimaan pengiriman. Konfirmasi pembayaran dilakukan ketika anggota telah melakukan transfer pembayaran sesuai dengan pesanan, dengan masukan input nama bank, nama pemilik rekening dan jumlah yang dibayarkan. Terlihat pada gambar 4.16

Sedangkan konfirmasi penerimaan pengiriman dilakukan sewaktu anggota telah menerima barang pesannya.

4.2.2.11 Halaman Edit Profil Anggota

Halaman edit anggota dapat merubah data diri anggota seperti perubahan alamat atau menambahkan alamat yang berbeda. Tampilan halaman bantuan terlihat pada lampiran gambar 4.17

4.2.2.12 Halaman Testimonials

Halaman testimonials *user* dapat anggotakan kesan/pesana mereka selama berbelanja di barbequa. Tampilan halaman testimonials terlihat pada gambar 4.18

4.2.2.13 Halaman Bantuan

Halaman bantuan merupakan menu bantuan kepada *user* dalam menjalankan aplikasi *mobile commerce* ini. Pada halaman ini menampilkan bantuan tata cara seperti, cara berbelanja, cara pembayaran dan cara konfirmasi pembayaran yang ditujukan kepada *user* yang masih baru dalam menjalankan aplikasi *mobile commerce* ini. Sehingga diharapkan *user* mendapatkan kemudahan dan mendapatkan informasi yang cukup pada sistem ini. Tampilan halaman bantuan terlihat pada lampiran gambar 4.19

4.2.3 Analisis Sistem

Analisa sistem adalah proses pengujian sistem yang bertujuan untuk mengetahui kinerja sistem terhadap kebenaran data pada sistem. Pengujian dilakukan pada setiap fungsi yang ada pada aplikasi dengan cara memasukkan data-data yang tidak valid atau kosong agar dapat diketahui bagaimana sistem dalam menangani data-data yang tidak valid atau kesalahan-kesalahan yang mungkin dilakukan oleh pengguna sistem. Materi pengujian meliputi komponen *input*, proses dan *output*.

4.2.3.1 Pengujian Kesalahan Deteksi Jaringan

Pada pengujian ini sistem aplikasi melakukan pemeriksaan perangkat jaringan pada *mobile devices*. Pemeriksaan ini bertujuan untuk penanganan kesalahan aplikasi sehingga aplikasi berjalan dengan baik, dikarenakan untuk menjalankan aplikasi *mobile commerce* ini diperlukan sebuah jaringan Internet. Apabila perangkat *blackberry user* tidak terdeteksi jaringan atau tidak tersambung jaringan, maka sistem akan memberikan informasi berupa kotak dialog informasi kepada *user*. Tampilan gambar penanganan kesalahan deteksi jaringan terlihat pada lampiran gambar 4.20

4.2.3.2 Pengujian Kesalahan Penambahan Produk

Penanganan kesalahan penambahan produk yang dimaksud adalah pemesanan *duplicate* yang dilakukan *user* pada pemilihan produk. Ketika *user* telah memilih produk untuk dipesan, kemudian *user* memilih produk yang sama, maka sistem akan memberikan informasi kepada *user* (lampiran gambar 4.21). Hal ini bertujuan agar sistem tidak mendapatkan produk yang sama dalam pemesanan, yang sebenarnya hal tersebut dapat dilakukan dengan perubahan kuantitas pada produk yang sama pada pesanan.

4.2.3.3 Pengujian Kesalahan Proses Registrasi

Pada pengujian ini sistem akan melakukan pengecekan form seperti pengecekan input kosong, pengecekan *username* dan pengecekan email. Pengujian ini bertujuan agar sistem mendapatkan informasi data *user* secara

detail, karena data tersebut sangat dibutuhkan yang nantinya digunakan untuk melakukan pengiriman pesanan. Selain itu pengecekan terhadap *username* dan email juga dilakukan dikarenakan *username* dan email merupakan data yang *unique* atau tidak boleh sama. Tampilan penanganan kesalahan proses registrasi terlihat pada lampiran gambar 4.22

4.2.4 Pengujian Sistem terhadap Admin

Pada pengujian ini dilakukan pada halaman admin, dimana admin dapat melakukan pengelolaan data. Pengujian admin akan dilakukan pada dua perangkat yang berbeda yakni perangkat PC (web) dan perangkat *mobile*.

4.2.4.1 Halaman Menu Admin

Halaman menu admin menampilkan informasi terkini dalam kuantitas seperti pesanan terbaru, anggota baru, testimonial baru dan total pemasukan. Pada halaman ini admin dapat mengetahui informasi terbaru, sehingga admin bisa memantau informasi yang datang setiap harinya. Pada perangkat PC terlihat seperti pada lampiran gambar 4.23.1 dan melalui perangkat *mobile* terlihat seperti pada lampiran gambar 4.23.2

4.2.4.2 Halaman Kelola Produk

Halaman kelola produk seorang admin dapat mengolah data produk, seperti tambah produk, edit produk dan tambah gambar produk. Pengujian halaman kelola produk ini hanya dilakukan pada perangkat *mobile*.

a. Menu Tambah Produk

Pada menu ini admin dapat melakukan penambahan produk dengan mengisi form yang dibutuhkan seperti kategori, harga, berat, ukuran, stok, deskripsi dan gambar. Tampilan halaman menu tambah produk terlihat pada lampiran gambar 4.24.1

b. Menu Edit Produk

Pada menu ini admin dapat mengubah informasi produk seperti mengubah kategori, nama, harga, stok, ukuran, berat dan deskripsi

produk. Tampilan halaman menu edit produk terlihat pada gambar 4.24.2

c. Menu Tambah Gambar Produk

Pada menu ini admin dapat menambahkan gambar produk Tampilan halaman menu edit produk terlihat pada lampiran gambar 4.24.3

4.2.4.3 Halaman Pesanan *User*

Halaman ini admin dapat melihat, dan merubah status pesanan *user*. Pengujian halaman pesanan *user* dilakukan dengan dua perangkat, yakni perangkat *mobile* dan web. Gambar lampiran 4.25.1 dan 4.25.2 menampilkan halaman pesanan *user* melalui perangkat *mobile* dan lampiran gambar 4.25.3 dan 4.25.4 pada web.

4.3 Hasil Implementasi Sistem

Setelah dilakukan implementasi sistem aplikasi *mobile commerce* dengan melakukan analisa dan pengujian sistem, dapat ditarik beberapa hasil yang dicapai sebagai berikut:

1. Aplikasi *mobile commerce* yang diimplementasikan dapat melakukan pertukaran data dengan sistem *server* melalui *web service*.
2. Aplikasi *mobile commerce* menggunakan media simpan data (*Persistent store*) untuk menyimpan hasil data dari *server* melalui *web service*, khususnya data-data produk.

4.3.1 Kelebihan Aplikasi

Aplikasi *mobile commerce* yang dibangun memiliki beberapa kelebihan, diantaranya :

- a. Sistem aplikasi *mobile commerce* ini tidak hanya diperuntukan untuk pengguna/pembeli saja, namun juga dapat dipakai oleh si pemilik atau *administrator* untuk mengelolanya.
- b. Dikarenakan aplikasi ini menggunakan media simpan data (*Persistent store*), pengguna dapat mengaksesnya dalam mode *offline* (tidak

terhubung jaringan).Tetapi dengan syarat pernah mengakses aplikasi ini sebelumnya.

4.3.2 Kekurangan Aplikasi

Namun aplikasi ini memiliki beberapa kelemahan terkait penggunaan media simpan data (persistent store) pada aplikasi, diantaranya adalah :

- a. Tidak semua menu dapat diakses dalam mode *offline*, artinya aplikasi tersebut hanya menampilkan katalog produk saja.
- b. Dampak perubahan terhadap data-data produk yang diakses secara *offline*. Jika pengguna sudah terlalu lama mengakses aplikasi *offline*, mungkin data-data tersebut tidak *terupdate*, seperti harga atau kuantitas yang mengalami perubahan.



BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan dari tugas akhir ini adalah sebagai berikut :

- a. Aplikasi *mobile commerce* yang dikembangkan memungkinkan proses perdagangan dilakukan secara *online* melalui perangkat nirkabel *smartphone* Blackberry. Transaksi penjualan dan pembelian produk yang dilakukan melalui perangkat *mobile*, sehingga transaksi bisa dilakukan kapan saja dan dimana saja.
- b. Pengembangan aplikasi *mobile commerce* memanfaatkan teknologi *web service* yang memungkinkan pertukaran data dengan perbedaan perangkat dan bahasa pemrograman (lintas platform), yang terjadi pada sisi *server* dengan aplikasi berbasis web dan sisi *client* pada perangkat *mobile*.
- c. Pada aplikasi *mobile commerce* ini menggunakan dua jenis format *web service*, yaitu JSON dan SOAP. Sehingga pertukaran data yang dilakukan disesuaikan berdasarkan aktifitasnya.
- d. Sistem aplikasi *mobile commerce* ini tidak hanya diperuntukan untuk pengguna/pembeli saja, namun juga dapat digunakan oleh si pemilik atau *administrator* untuk mengelolanya.
- e. Sistem aplikasi *mobile commerce* ini dapat digunakan dalam mode luring (*offline*).Dimanapengguna dapat mengaksesnya walaupun tidak terhubung atau terkoneksi jaringan, tetapi dengan syarat pernah mengakses aplikasi ini sebelumnya.

5.2 Saran

Aplikasi ini masih jauh dari sempurna, karena itu diharapkan di waktu yang mendatang bisa diperbaiki dan disempurnakan. Adapun saran-saran yang dapat diberikan terhadap aplikasi ini adalah:

- a. Pada aplikasi ini desain tampilan yang diberikan masih bersifat sederhana, diharapkan lebih ditingkatkan dalam segi tampilan seperti penggunaan latar gambar atau menambahkan video jika diperlukan, yang bertujuan menarik para konsumen dalam menggunakan aplikasi ini.
- b. Dengan memanfaatkan teknologi *web service* aplikasi *mobile commerce* ini dapat dikembangkan tidak hanya diperuntukan untuk Blackberry saja, melainkan untuk *smartphone* yang lain seperti Android yang semakin populer belakangan ini.
- c. Menyediakan layanan untuk memberitahukan posisi keberadaan *user* dengan layanan berbasis lokasi (*Location Based Service*) yang menggunakan teknologi GPS (*Global Positioning System*), sehingga dapat mengantarkan *user*/pembeli menuju toko dengan mudah melalui denah (Blackberry maps/Google maps).
- d. Sistem pembayaran pada aplikasi ini masih menggunakan transfer melalui ATM. Diharapkan kedepan menyediakan layanan transfer uang melalui elektronik seperti paypal, sehingga memberikan pelayanan yang memudahkan *user*/pembeli dalam pemesanan.

DAFTAR PUSTAKA

Lucky.2008. XML Web Service : Aplikasi Desktop, Internet, Handphone. Jakarta: Jasakom

Budi Raharjo, 2007. Tuntunan Pemrograman JAVA untuk Handphone. Bandung: Informatika

Anthony Rizk, 2009. Beggining Blackberry Development, Appress



LAMPIRAN

Gambar Web Service

```
- products: [
  - {
    id: 10,
    category: "Pantofel",
    product_name: "Clark Kuning",
    description: "<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>",
    price: 400000,
    img: http://barbequa.nitelab.web.id/utility/image/?id=13&w=100&h=75&hash=65460ec088f62bc062805d07a0648ae1,
    comments: {
      count: 0
    },
    ratings: 0,
    publish: "16 May 2011"
  },
  - {
    id: 9,
    category: "Boots",
    product_name: "Dr.Martin",
    description: "<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>",
    price: 450000,
    img: http://barbequa.nitelab.web.id/utility/image/?id=12&w=100&h=75&hash=7c418bba4c61b0ee03cc266cce6450ca,
    comments: {
      count: 0
    },
    ratings: 0,
    publish: "16 May 2011"
  },
  - {
    id: 8,
    category: "Pantofel",
    product_name: "Bolle Perempuan",
    description: "<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>",
    price: 350000,
    img: http://barbequa.nitelab.web.id/utility/image/?id=11&w=100&h=75&hash=9d82626ba298394c0d5f15c393cbe694,
    comments: {
      count: 0
    }
  }
]
```

Gambar 4.1 JSON Daftar Produk

```
--- --
category: "Pantofel",
product_name: "E shoe",
stock: 7,
description: "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.",
price: 460000,
img: http://bbqsf/utility/image/?id=13&w=200&h=150&hash=c639039842468aa5fd410fc83c03c656,
comments: {
  - data: [
    - {
      name: "Admin",
      email: "0aeebd0ba59e9ccc45f751d7d7fe07fa",
      comment: "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua",
      created: "28 Apr 2011"
    },
    - {
      name: "Zss",
      email: "e98bfef3cd27c11023fdcc048fe63fb7",
      comment: "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua",
      created: "25 Apr 2011"
    },
    - {
      name: "Dani",
      email: "cd9db5a678b4def797459e3e5fe3be27",
      comment: "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua",
      created: "25 Apr 2011"
    }
  ],
  count: 3
},
ratings: 5,
publish: "01 Apr 2011"
```

Gambar 4.2 JSON Detail Produk

```

- data: [
  - {
    id_kota: 108,
    provinsi: "D.I.Aceh",
    nama_kota: "banda aceh",
    tarif: 25500
  },
  - {
    id_kota: 109,
    provinsi: "Sumatera Utara",
    nama_kota: "medan",
    tarif: 21000
  },
  - {
    id_kota: 110,
    provinsi: "Riau",
    nama_kota: "pekanbaru",
    tarif: 19000
  },
  - {
    id_kota: 111,
    provinsi: "Kepulauan Riau",
    nama_kota: "batam",
    tarif: 18500
  }
]

```

Gambar 4.3 JSON Tarif

```

{
  first_name: "test",
  last_name: "soap",
  email: "test@soap",
  no_telp: "123456789",
  alamat: "jln kaliurang km 14",
  alamat2: "",
  - kota: {
    id_kota: 53,
    nama_kota: "pekalongan",
    tarif: 17000
  },
  kode_pos: "53224"
}

```

Gambar 4.4.1 JSON Informasi User

```
- Exception: {
  type: "AuthorizationException",
  message: "An active access token must be used to query information about the current user"
}
```

Gambar 4.4.2 Exception JSON Informasi *user*

```
- data: [
  - {
    id: 1,
    no_order: "B530Q0",
    - orders: {
      - data: [
        - {
          id: 6,
          product: "Sally 42",
          img: http://barbequa.nitelab.web.id/utility/image?id=9&w=100&h=75&service=true,
          price: 385000,
          qty: 1
        },
        - {
          id: 7,
          product: "Sally Coklat",
          img: http://barbequa.nitelab.web.id/utility/image?id=10&w=100&h=75&service=true,
          price: 420000,
          qty: 2
        }
      ]
    },
    total: 1305000,
    status: 2,
    date: "01 Jun 2011"
  }
]
```

Gambar 4.5.1 JSON Pemesanan *User*

```
- Exception: {
  type: "AuthorizationException",
  message: "An active access token must be used to query information about the current user"
}
```

Gambar 4.5.2 Exception JSON Pemesanan *User*

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

- <definitions name="Soap_Service" targetNamespace="http://barbequa.nitelab.web.id/api/soap">
- <types>
- <xsd:schema targetNamespace="http://barbequa.nitelab.web.id/api/soap">
- <xsd:complexType name="ArrayOfProduct">
- <xsd:complexContent>
- <xsd:restriction base="soap-enc:Array">
<xsd:attribute ref="soap-enc:arrayType" wsdl:arrayType="tns:Product[]"/>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
- <xsd:complexType name="Product">
- <xsd:all>
<xsd:element name="id" type="xsd:string"/>
<xsd:element name="product_name" type="xsd:string"/>
<xsd:element name="price" type="xsd:string"/>
<xsd:element name="img" type="xsd:string"/>
<xsd:element name="comments" type="xsd:string"/>
<xsd:element name="ratings" type="xsd:string"/>
<xsd:element name="publish" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
- <xsd:complexType name="Order">
- <xsd:all>
<xsd:element name="id" type="xsd:string"/>
<xsd:element name="code" type="xsd:string"/>
<xsd:element name="total" type="xsd:string"/>
<xsd:element name="phone" type="xsd:string"/>
<xsd:element name="transfer" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
- <xsd:complexType name="Action">
- <xsd:all>
<xsd:element name="action" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
</types>
</definitions>

```

Gambar 4.6.3 SOAP WSDL

The screenshot shows a web service client interface with the following details:

- WSDL Address:** http://barbequa.nitelab.web.id/api/soap/wsdl
- Communication:** async (selected)
- Service name:** Soap_ServiceService
- Description:** Konfirmasi penerimaan pesanan
- Targetnamespace:** http://barbequa.nitelab.web.id/api/soap
- Operations:**
 - 1. Operation: getProducts
 - 2. Operation: setNewOrder
- username:** (input field)
- (type):** xsd:string
- Raw WSDL Data:**

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://barbequa.nitelab.web.id/api/soap" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" name="Soap_Service" targetNamespace="http://barbequa.nitelab.web.id/api/soap">
<types>
<xsd:schema targetNamespace="http://barbequa.nitelab.web.id/api/soap">
<xsd:complexType name="ArrayOfProduct">
<xsd:complexContent>
<xsd:restriction base="soap-enc:Array">
<xsd:attribute ref="soap-enc:arrayType" wsdl:arrayType="tns:Product[]"/>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Product">
<xsd:all>
<xsd:element name="id" type="xsd:string"/>
<xsd:element name="product_name" type="xsd:string"/>
<xsd:element name="price" type="xsd:string"/>
<xsd:element name="img" type="xsd:string"/>
<xsd:element name="comments" type="xsd:string"/>
<xsd:element name="ratings" type="xsd:string"/>
<xsd:element name="publish" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
<xsd:complexType name="Order">
<xsd:all>
<xsd:element name="id" type="xsd:string"/>
<xsd:element name="code" type="xsd:string"/>
<xsd:element name="total" type="xsd:string"/>
<xsd:element name="phone" type="xsd:string"/>
<xsd:element name="transfer" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
<xsd:complexType name="Action">
<xsd:all>
<xsd:element name="action" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
</types>
</definitions>

```
- Status:** 200 OK

Gambar 4.6.3 Service Method dengan SOA Client

Gambar Pengujian Sistem User



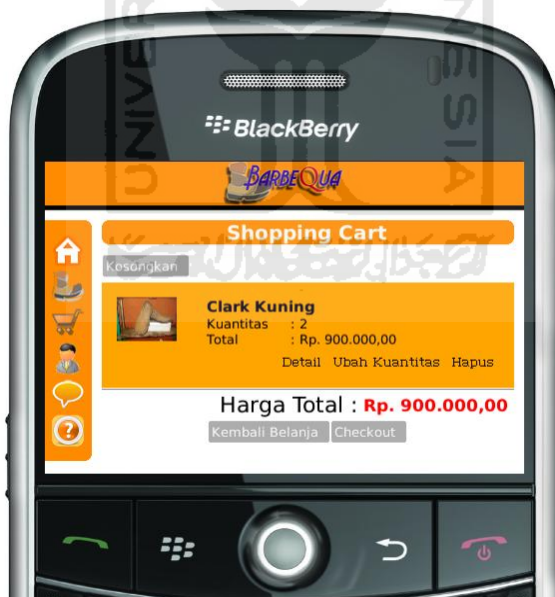
Gambar 4.7 Halaman Utama



Gambar 4.8 Halaman Daftar Produk



Gambar 4.9 Halaman Detail Produk



Gambar 4.10 Halaman Keranjang Belanja



Gambar 4.11 Halaman *Checkout*Akun



Gambar 4.12 Halaman *Checkout*Registrasi



Gambar 4.13.1 Halaman *Checkout* Login Anggota



Gambar 4.13.2 Halaman *Checkout* sebagai Anggota



Gambar 4.14 Halaman *Checkout* Selesai



Gambar 4.15.1 Halaman Akun Login Anggota



Gambar 4.15.2 Halaman Akun Anggota



Gambar 4.16 Halaman Konfirmasi Pembayaran Pesanan Anggota



Gambar 4.17 Halaman Edit Profil Anggota



Gambar 4.18 Halaman Testimonials



Gambar 4.19 Halaman Bantuan

Gambar Analisis Sistem



Gambar 4.20 Halaman Deteksi Jaringan



Gambar 4.21 Halaman Kesalahan Penambahan Produk



Gambar 4.22 Halaman Kesalahan Proses Registrasi

Gambar Pengujian Sistem Admin

Dashboard Category Product Customer Order Profile Setting Logout

Home > Dashboard

Right Now [+ Add New Product](#)

You have **2 new orders**, **0 new users** and **0 new reviews**, Total of **Rp 1.305.000,- profit**

Last 5 Orders

Customer	Items	Grand Total
Test Soap	1	Rp 817.000,-
Dani Gojay	1	Rp 812.000,-

Bestsellers

Product Name	Price	Orders
Clark Kuning	Rp 400.000,-	4
Bally Coklat	Rp 420.000,-	2
Bally 42	Rp 385.000,-	1

Last Reviews

Author	Comment
No comments	

Gambar 4.23.1 Halaman Menu Admin pada Web



Gambar 4.23.2 Halaman Menu Admin pada *Mobile*



Gambar 4.24.1 Halaman Menu Tambah Produk pada *Mobile*



Gambar 4.24.2 Halaman Menu Edit Produk pada *Mobile*



Gambar 4.24.3 Halaman Menu Tambah Gambar Produk pada Mobile



Gambar 4.25.1 Halaman Menu Lihat Pesanan *User* pada *Mobile*



Gambar 4.25.2 Halaman Menu Ubah Pesanan *User* pada *Mobile*

Dashboard Category Product Customer **Order** Profile Setting Logout

Home > Manage Order

i There are currently 3 orders.

Id	Status	Transaction	Customer	Total	Created	Options
3	New	B608TS	Test Soap	Rp 817.000,-	08 June 2011	view delete
2	New	B608DG	Dani Gojay	Rp 812.000,-	08 June 2011	view delete
1	Paid	B530QU	Test Soap	Rp 1.305.000,-	30 May 2011	view delete

Gambar 4.25.3 Halaman Lihat Pesanan *User* pada *WebServer*

Dashboard Category Product Customer **Order** Profile Setting Logout

Home > Manage Order > View Order

Transaction : B608TS Order Created : 09 June 2011 11:56:10

Status

JNE Tracking

Order Payment on Jun 9, 2011 9:35:29 PM

Shipping Address

Name	: Test Soap
Address	: Jln Kaliurang Km 14
City	: Pekalongan
Province	: Jawa Tengah
Postal Code	: 53224
Phone	: 123456789

Pic	Product	Unit Price	Quantity
	Clark Kuning	Rp 400.000,-	2
Total			Rp 817.000,-

Gambar 4.25.4 Halaman Ubah Pesanan *User* pada *WebServer*