

## PENGEMBANGAN PERANGKAT LUNAK AKSES *PUSH MAIL* VIA YAHOO MESSENGER

Muhammad Aditya Nugraha<sup>1</sup>, Gede Karya<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Informatika, Fakultas Teknologi Informasi dan Sains,  
Universitas Katolik Parahyangan, Bandung – 40141  
E-mail : adityanugraha14@yahoo.com<sup>1</sup>, gkarya@unpar.ac.id<sup>2</sup>

### ABSTRAK

Makalah ini membahas tentang pengembangan aplikasi akses *Push Mail* (PM) menggunakan *Yahoo Messenger* (YM). Pengembangan perangkat lunak ini terinspirasi oleh banyaknya layanan paket data murah yang disediakan oleh operator seluler khusus untuk chatting, sementara layanan PM masuk ke dalam paket yang lebih mahal. Aplikasi yang dikembangkan bersifat generik, atas dasar layanan email yang menggunakan protokol *Simple Mail Transfer Protocol* (SMTP), *Post Office Protocol version 3* (POP3) dan *Internet Message Application Protocol* (IMAP). Komunikasi dengan YM server menggunakan YM SDK (*Yahoo Messenger Software Development Kit*). Aplikasi dibangun dengan menggunakan bahasa pemrograman Java pada lingkungan *Java 2 Standar Edition* (J2SE). Sedangkan di sisi client, menggunakan aplikasi standar YM Client yang sudah ada. Dengan demikian, semua perangkat yang dilengkapi dengan YM Client dapat digunakan untuk mengakses layanan aplikasi dengan biaya murah.

**Kata kunci :** *Push Mail, Yahoo Messenger*

### 1. PENDAHULUAN

*Push Mail* (PM) merupakan layanan pengantaran email secara otomatis, sehingga pengguna tidak perlu mengambil sendiri email-nya ke server.

Layanan operator ponsel saat ini banyak memberikan paket data dengan segmentasi pasar beraneka ragam. Khusus untuk layanan chatting lebih murah dari pada paket PM yang biasanya ada di paket *full service*. Sebagai contoh, 3 operator besar di Indonesia, yaitu: Telkomsel memberikan paket *Social Networking & Chatting* (SNC) Rp. 20.000,-/ bulan [1], sementara Indosat memberikan layanan sejenis Rp. 45.000,-/bulan [2] dan XL Rp. 50.000,- / bulan [3]. Sementara layanan *full service* Rp. 99.000,- ke atas. Layanan PM didominasi oleh Blackberry dengan model layanan yang private menjadi suatu tantangan untuk mengembangkan layanan PM yang handal dan murah.

Oleh karena itu, dalam penelitian ini akan dibahas mengenai pengembangan aplikasi PM yang berjalan di atas layanan *Yahoo! Messenger* (YM) sebagai transport. Perangkat lunak dikembangkan atas dasar model integrasi PM dengan *Instance Messenger* (IM) pada makalah sebelumnya [4], dengan penyesuaian pada protokol YM yang proprietary. Perangkat lunak diimplementasikan menggunakan bahasa pemrograman Java pada lingkungan *Java 2 Standar Edition* (J2SE). Pemilihan YM didasarkan atas realita dan asumsi, bahwa semua perangkat mobile yang menyediakan akses ke internet dapat menggunakan layanan ini. Selain itu, sifatnya yang interaktif, banyaknya pengguna dan biaya akses layanan yang murah yang disediakan oleh semua operator selular di Indonesia.

Berdasarkan uraian di atas, maka masalah yang akan dibahas dalam makalah ini adalah: (1) Bagaimana konsep dan cara kerja aplikasi *Yahoo Messenger* API? (2) Bagaimana mengintegrasikan layanan *Push Mail* menggunakan layanan *Yahoo Messenger*? Adapun kontribusi yang diharapkan adalah: dengan adanya aplikasi ini para pengguna perangkat mobile dapat mengakses layanan PM dengan tarif murah, yaitu tarif YM (paket chatting atau SNC atau gaul dan sejenisnya).

### 2. ANALISA, DESAIN, DAN IMPLEMENTASI

Pada bagian ini akan dibahas kajian pustaka tentang layanan email, PM dan YM SDK. Kemudian dilanjutkan dengan pembahasan integrasi layanan PM menggunakan protokol YM.

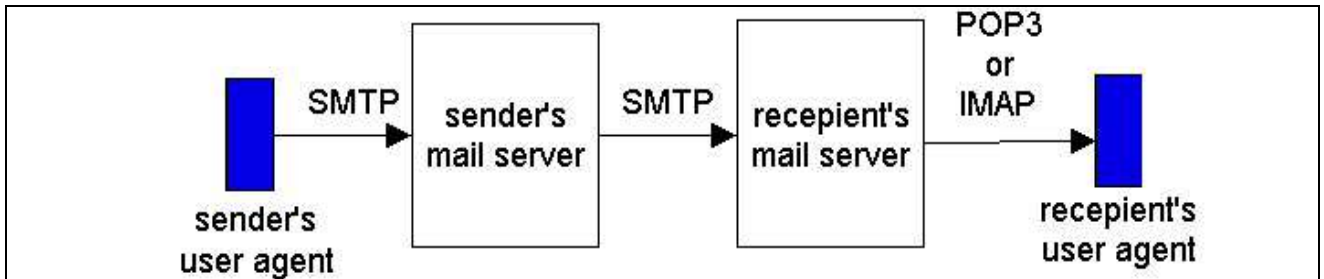
#### 2.1. Layanan Email

Email merupakan surat yang dikirim secara elektronik. Struktur email terdiri atas alamat email pengirim (*from*), alamat email tujuan (*to*, *cc*, *bcc*), *subject*, *body* dan atribut *header* lainnya. Struktur email diatur dalam RFC 5322 [5]. Protokol yang digunakan untuk mengirim email adalah *Simple Mail Transfer Protocol* (SMTP) [6]. Sedangkan untuk mengakses

email dari email server oleh pengguna menggunakan protokol *Post Office Protocol* versi 3 (POP3) [7] atau menggunakan *Internet Message Access Protocol* (IMAP) [8].

Untuk meningkatkan keamanan, baik pengiriman maupun penerimaan/ pengaksesan email dapat ditambahkan dengan modul *Secure Socket Layer* (SSL) pada level transport. Selain itu pengamanan juga dilakukan melalui prosedur *authentication* dan *autorization* menggunakan *user id* dan *password*.

Dari sisi fitur, layanan email minimal menyediakan fitur-fitur: (1) Pendaftaran akun email, (2) Autentifikasi dan otorisasi, (3) Pembuatan/ komposisi surat, (4) Pengiriman surat ke tujuan, (5) pengambilan email dari server email, (6) pembacaan dan tindak lanjut berupa *reply* atau *forward*, (7) pengelompokan/ filtering dan pengarsipan. Mekanisme pengiriman dan penerimaan *email* lebih jauh dapat dilihat pada gambar 1.



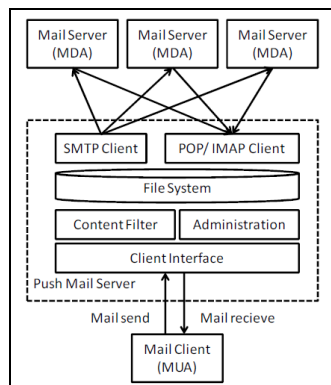
Gambar 1: Sistem Kerja *Email*

Pada gambar 1, menjelaskan sistem kerja *email* mulai dari dikirimkan hingga diambil oleh penerima. Pertama-tama *user agent* pengirim mengirimkan pesan kepada *mail server* milik pengirim. *Mail server* pengirim kemudian memeriksa tujuan *email* tersebut dan meneruskannya kepada *mail server* penerima. *Email* tersebut disimpan oleh *mail server* penerima hingga pengguna mengambil sendiri pesan tersebut dengan menggunakan *user agent*-nya.

## 2.2. Layanan *Push Mail*

Seperti dijelaskan pada bagian 1, bahwa *Push Mail* (PM) merupakan layanan pengantaran email secara otomatis, sehingga pengguna email tidak perlu mengambil sendiri emailnya ke server. Dengan menggunakan layanan BlackBerry sebagai referensi [4][8], maka dapat diidentifikasi beberapa fitur dari layanan PM, antar lain: (1) Pendaftaran akun email yang ingin di-*push*, (2) *Content filtering* email yang akan di-*push*, (3) mengambil email dari MDA, dan menyimpannya di PM Server secara periodik, (4) mengirimkan (mempush) email dari PM Server ke MUA sesuai dengan kriteria filter, (5) melakukan sinkronisasi akses dari MUA seperti: penghapusan dan sejenisnya, (6) mengambil email yang akan dikirim oleh MUA, (7) mengirimkan email ke MDA, (8) pengelolaan *file attachment*, baik email datang maupun email yang dikirim. Interaksi antara PM Server, MDA dan MUA dapat dilihat pada gambar 2.

Pada gambar 2, dapat dilihat bahwa MUA tidak berinteraksi langsung dengan MDA, melainkan ke PM Server melalui modul *Client Interface*. PM Server-lah yang bertugas untuk mengambil email dari MDA menggunakan POP/IMAP Client dan mensinkronisasi ke MUA sesuai dengan aturan yang didefinisikan melalui *Content filtering*. Demikian juga pengiriman email, dilakukan oleh PM Server melalui modul SMTP Client. Dengan model ini juga dimungkinkan PM Server mengelola dan mensinkronisasi beberapa akun email dari MDA yang berbeda untuk 1 MUA.



Gambar 2 : Interaksi PM Server, MDA, dan MUA

### 2.3. Yahoo Messenger SDK

Yahoo Messenger SDK (YM SDK) merupakan development tools resmi milik Yahoo untuk membangun aplikasi Yahoo Messenger client. YM SDK berbasis HTTP dan mengatur format request dan response untuk mengakses server YM. Format yang diatur meliputi : alamat URL request, data paramater, data yang dipertukarkan, dan metode request pada HTTP seperti GET, POST, PUT, atau DELETE. YM SDK diakses dengan langkah-langkah sebagai berikut :

1. Mengakses situs "http://developer.yahoo.com" untuk mendapatkan Open Authentication(OAuth) key yang akan digunakan oleh aplikasi untuk mengakses server YM. Oauth key terdiri dari dua data utama: Oauth consumer key dan consumer secret key.
2. Melakukan authentication kepada server YM dengan menggunakan protokol Direct OAuth. Bagian ini dilakukan oleh aplikasi YM client yang dibangun.
3. Aplikasi mengakses YM API (Application Programming Interface) untuk menggunakan layanan yang disediakan oleh YM SDK seperti : login, mengirim pesan, logout, dll.

Terdapat banyak layanan yang disediakan oleh YM API untuk membangun YM client seperti session management, message management, buddy list authorization management, dll. Contoh format request untuk mengirimkan pesan dapat dilihat pada tabel 2[9].

Tabel 1: Format Message Management API

HTTP	Fungsi	Input URL	Request Body (JSON)	Response Body (JSON)
POST	Mengirim pesan	sid	Message sendAs (optional)	<Kosong>

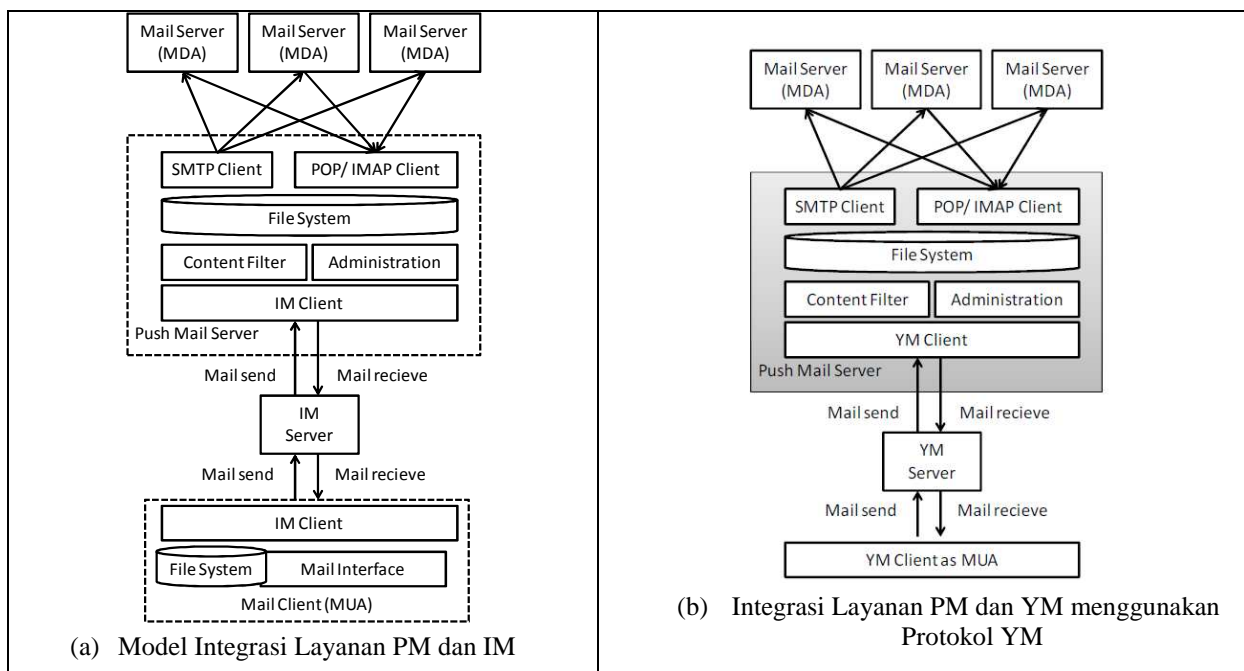
Definisi dari setiap data yang dipertukarkan pada *Message Management* API dapat dilihat pada tabel 2 [9].

Tabel 2 : Definisi data Message Management API

Atribut	Tipe Data	Deskripsi
Message	String	Pesan yang akan dikirimkan.
sendAs	String	Berisi <i>id</i> Yahoo yang digunakan untuk mengirim pesan.
Sid	String	Berisi <i>session id</i> yang didapatkan ketika <i>login</i> .

### 2.4. Integrasi PM dengan YM

Pada makalah sebelumnya [4] telah diusulkan model integrasi PM dengan IM seperti pada gambar 3.



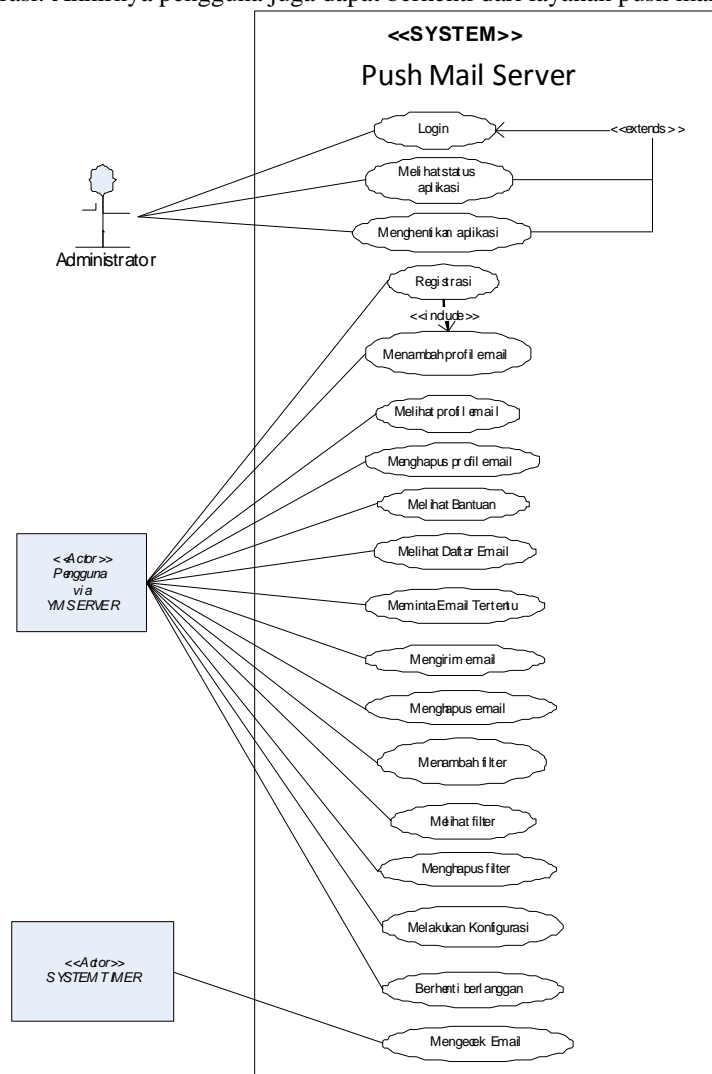
Gambar 3: Model Integrasi Layanan PM dan YM

Pada gambar 3 (a) dapat dilihat bahwa pesan-pesan email yang tadinya langsung dikomunikasi oleh PM Server dengan MUA, dengan model ini dilewatkan melalui IM Server. Dengan demikian tidak ada komunikasi lain di luar protokol IM. Untuk itu, *Client Interface* di PM Server diganti dengan IM Client yang menggunakan protokol IM. Sedangkan di sisi MUA juga antarmukanya diganti dengan IM Client.

Dengan menggantikan IM dengan YM, maka model tersebut dapat disesuaikan menjadi seperti pada gambar 3 (b). Khusus peran MUA di sisi client, pada makalah ini menggunakan sepenuhnya client dari YM. Aplikasi sesuai model 3 (b) diberi nama Push Mail Server. *Use Case* dari aplikasi Push Mail Server dapat dilihat pada gambar 4.

Pada gambar 4 dapat dilihat bahwa ada 3 aktor yang berperan dalam sistem ini, yaitu:

1. **Administrator**, yang bertugas untuk mengelola aplikasi push mail ini. Aksi yang dapat dilakukan adalah: login, melihat status aplikasi server apakah masih berjalan atau tidak, dan menghentikan aplikasi push mail server.
2. **Pengguna**. Melalui YM Server, dengan akun YM yang dimiliki dapat melakukan registrasi, menambah profil berupa akun email yang akan di push berikut server SMTP, POP3/IMAP dan userid serta password yang dapat digunakan untuk mengakses akun email tersebut. Selain menambah profil, juga dapat melihat dan menghapus profilnya. Untuk setiap pengguna dapat memiliki lebih dari 1 profil email. Jika diperlukan, pengguna dapat mengirimkan perintah untuk menampilkan bantuan. Untuk membaca dan mengirim email, pengguna dapat melakukan aksi melihat daftar email, meminta email tertentu, mengirim email baru dan menghapus email yang sudah ada. Selain itu juga ada fungsi filter yang dapat diakses oleh pengguna dengan aksi menambah, melihat dan menghapus filter email tertentu. Untuk mengatur frekuensi waktu pengecekan dan banyaknya email yang dapat ditampilkan, maka pengguna dapat melakukan aksi melakukan konfigurasi. Akhirnya pengguna juga dapat berhenti dari layanan push mail (*unregister*).



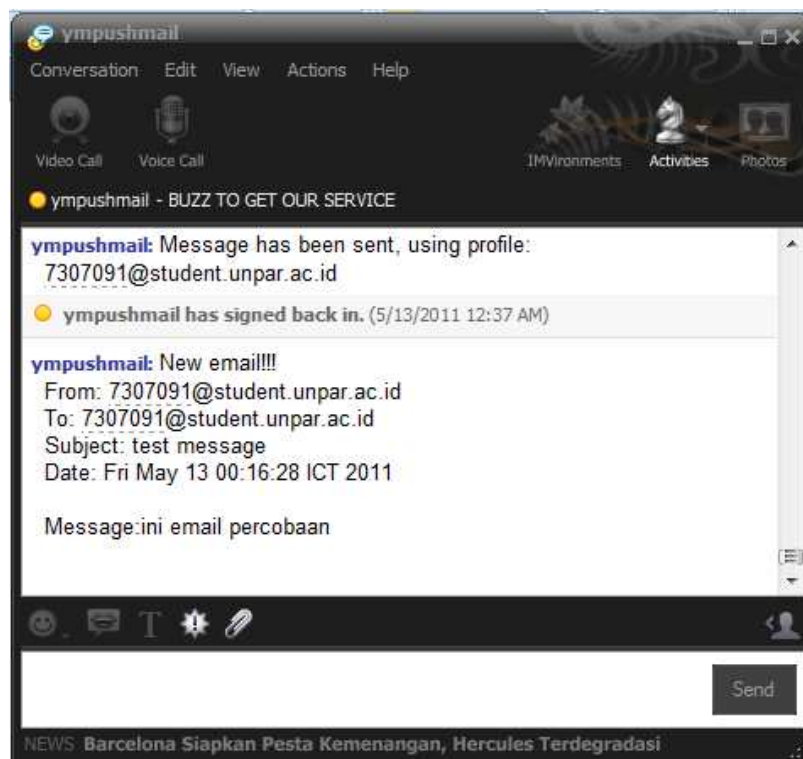
Gambar 4: Use Case Aplikasi YM Push Mail

3. **System Timer.** Merupakan aktor yang akan menjalankan aksi mengecek email sesuai dengan konfigurasi yang dibuat oleh Pengguna.

Aplikasi Push Mail Server (pada gambar 4) telah dikembangkan dengan bahasa Java pada lingkungan *Java 2 Standar Edition (J2SE)*. Untuk akses ke YM Server digunakan *YM Application Programming Interface (YM API)* yang merupakan bagian dari *YM Software Development Kit (SDK)*. Dengan implementasi seperti gambar 3(b), maka semua perangkat yang dapat menjalankan YM Client dapat menikmati layanan Push Mail.

### 3. HASIL DAN PEMBAHASAN

YM PushMail memberikan layanan melalui akun YM tertentu, misalnya: *ympushmail*. Jika ingin memanfaatkan layanan ini, user harus telah memiliki akun YM, misalnya: *adt14*. Setelah itu login ke YM Server, user mengirim pesan `<buzz>` ke *ympushmail*. PM Server akan mengirimkan pesan-pesan yang dapat digunakan untuk: (1) mendaftarkan akun email yang akan di-*push*, (2) menentukan mode *push*, apakah *header* saja, atau termasuk *body*, (3) menentukan *content filter* email yang akan di-*push*, (4) perintah untuk mengambil email dengan nomor tertentu, jika modenya adalah *header*. Sebagai contoh uji dapat dilihat pada gambar 5.



Gambar 5: Contoh Hasil Uji

Pada gambar 5, setelah melakukan registrasi email, maka untuk setiap email yang masuk ke mailbox *adt14*, akan di-*push* jika akun YM *adt14* sedang *online*. Bagaimana jika sedang *off line*? Email tersebut akan dikirim sebagai *offline message* ke akun YM tersebut. Dengan model ini, tidak akan ada email yang hilang.

Berdasarkan uraian pada bagian 2 dan 3 maka masalah yang dikemukakan pada bagian 1 telah terjawab. Aplikasi dapat memberikan layanan PM kepada pengguna melalui pesan YM. Dengan bentuk tersebut memungkinkan pengguna mengakses layanan PM menggunakan berbagai perangkat yang mempunyai aplikasi YM *client*. Selain itu, pengguna juga dapat berpindah-pindah perangkat untuk mengakses layanan PM dari aplikasi yang tidak dapat dilakukan pada layanan PM lainnya. Pengguna kini dapat mengakses layanan PM dengan biaya paket yang lebih murah yang disediakan oleh operator. Bentuk aplikasi ini dapat diakses dari berbagai aplikasi YM *client*.

### 4. PENUTUP

Dari hasil pembahasan dapat disimpulkan integrasi layanan PM menggunakan layanan YM dapat dibangun dan berjalan dengan baik. Sehingga memungkinkan untuk mengakses layanan PM dengan biaya yang lebih murah.

Interaksi user dengan Push Mail Server dengan aplikasi YM Client menggunakan mode *command line interface* (CLI) kurang nyaman untuk user awam. Untuk itu perlu dikembangkan lebih lanjut MUA yang berbasis GUI (*Graphical User Interface*) sesuai dengan model 3 (a) sehingga lebih memudahkan pengguna untuk mengakses layanan PM dari aplikasi.

## DAFTAR REFERENSI

- [1] Telkomsel (2011) Paket Layanan Facebookdan Chatting (Social Networking &Chatting/SNC) Telkomsel[Online]. Available at: <http://www.telkomsel.com/mobile/productdetail-content/108/495> [Accessed: 10 Februari 2011].
- [2] Indosat (2011) BlackBerry(r) On Demand Gaul [Online]. Available at: [http://www.indosat.com/Matrix\\_BlackBerry/BlackBerry\\_Internet\\_Service/BlackBerryr\\_On\\_Demand\\_Gaul](http://www.indosat.com/Matrix_BlackBerry/BlackBerry_Internet_Service/BlackBerryr_On_Demand_Gaul) [Accessed: 10 Februari 2011].
- [3] XL Asiata (2011) BlackBerry Gaul [Online]. Available at: <http://www.xl.co.id/language/idid/blackberry/paketlayanan/bisgaul> [Accessed: 10 Februari 2011].
- [4] G. Karya., M. Aditya, 2012, "Model Pengembangan Aplikasi Push Mail Murah Menggunakan Protokol Instance Messaging", Prosiding pada Seminar Nasional Teknik Informatika (SANTIKA) 2012, Surabaya, 10 Maret 2012.
- [5] J. Klensin (2008) RFC 5321: Simple Mail Transfer Protocol. Network Working Group.
- [6] J. Myers, Carnegie Mellon, M. Rose (1996) RFC 1939: Post Office Protocol - Version 3. Network Working Group.
- [7] M. Crispin (2003) RFC 3501: Internet Message Access Protocol Version 4 rev 1. Network Working Group.
- [8] Indosat & Blackberry (2011) Blackberry Web Base Application [Online]. Available at: <https://indosat.blackberry.com> [Accessed: 10 Februari 2011].
- [9] Yahoo! (2010) Yahoo Messenger SDK [Online]. Available at: <http://developer.yahoo.com/messenger/guide> [Accessed: 19 September 2011].