

## Pemanfaatan Instant Messaging untuk Aplikasi Layanan Akademik

Eri Zuliarso dan Herny Februariyanti

Fakultas Teknologi Informasi, Universitas Stikubank  
email: eri299@gmail.com, herny@unisbank.ac.id

### Abstrak

Instant Messaging (IM) saat ini mengalami perkembangan yang cukup pesat pada jaringan user, karena kemampuannya mengirimkan pesan secara singkat dan cepat antara pengguna telekomunikasi. IM menjadi perangkat yang sangat penting untuk industri di seluruh dunia. IM digunakan di dalam penjadwalan (*scheduling meeting*), pertukaran informasi bisnis dan informasi client dan lain-lain. IM telah dikembangkan pada sektor- sektor private atau antar provider seperti American Online Instant Messenger (AIM), MSN dan Yahoo.

Teknologi informasi yang sudah berkembang memberikan trend penyebaran informasi bukan hanya memakai media spanduk maupun brosur, tetapi melalui Instant messaging. Hal ini dikarenakan computer, laptop, komputer tablet, telepon genggam sudah merupakan kebutuhan primer dan selalu melekat kemanapun pemiliknya pergi. Sehingga dengan dasar ini, peneliti mencoba mengimplementasikan aplikasi Instant messaging sebagai sarana penyebaran informasi ke mahasiswa dengan tujuan utama adalah tersampainya informasi-informasi dari kampus langsung kepada mahasiswa.

Penelitian ini dibangun Sistem Layanan Informasi Akademik menggunakan Instant Messenger. Instant Messenger yang digunakan memanfaatkan layanan Google Talk, untuk dapat memanfaatkan fasilitas ini pemakai harus mempunyai account email gmail (xxxx@gmail.com). Dalam penelitian ini sistem dapat menyebarkan informasi kepada seluruh mahasiswa, sistem dapat menjawab kebutuhan request informasi akademik dari civitas akademika dan sistem mempunyai kemampuan untuk mengauthentikasi sms dari civitas akademika. Sehingga informasi hanya dapat di akses oleh civitas akademika yang berhak.

**Kata kunci :** instant messanging, layanan informasi akademik

### PENDAHULUAN

*Instant Messaging* (IM) saat ini mengalami perkembangan yang cukup pesat pada jaringan *user*, karena kemampuannya mengirimkan pesan secara singkat dan cepat antara pengguna telekomunikasi. IM menjadi perangkat yang sangat penting untuk industri di seluruh dunia. IM digunakan di dalam penjadwalan (*scheduling meeting*), pertukaran informasi bisnis dan informasi *client* dan lain-lain. IM telah dikembangkan pada sektor- sektor *private* atau antar *provider* seperti *American Online Instant Messenger* (AIM), MSN dan Yahoo. Pada tahun 1998 muncul protokol IM yang bersifat *open source* yang terkenal dengan protokol Jabber.

Jabber mulai dapat perhatian publik ketika

didiskusikan antar *developer* pada *website* Slashdot pada bulan Januari 1999. Pada Mei 2000, protokol Jabber diluncurkan sebagai protokol yang bersifat *open source* berdasarkan referensi server dan saat ini tidak dapat saling dipertukarkan. Jabber menggunakan arsitektur *client-server*, bukan arsitektur *peer- to-peer* seperti yang digunakan pada sistem IM lainnya. Protokol Jabber menggunakan format pesan *Extensible Markup Language* (XML). Format dokumen XML mejadi bahasa generik yang digunakan pada berbagai aspek komunikasi, karena sifatnya yang berbasis teks, mudah dibaca oleh manusia, maka aplikasi yang berbasis XML mudah untuk *didebug* atau melewati *firewall*.

Teknologi informasi yang sudah berkembang memberikan trend penyebaran informasi bukan hanya memakai media spanduk maupun brosur, tetapi melalui Instant messaging. Hal ini dikarenakan computer, laptop, komputer tablet, telepon genggam sudah merupakan kebutuhan primer dan selalu melekat kemanapun pemiliknya pergi. Sehingga dengan dasar ini, peneliti mencoba mengimplementasikan aplikasi Instant messaging sebagai sarana penyebaran informasi ke mahasiswa dengan tujuan utama adalah tersampainya informasi-informasi dari kampus langsung kepada mahasiswa.

## METODE PENELITIAN

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah sebagai berikut:

### 1. Obyek Penelitian

Obyek penelitian dari penelitian ini adalah perpustakaan Universitas Stikubank Semarang.

### 2.. Data yang diperlukan

Merupakan data yang mendukung dalam penelitian ini meliputi data primer dan data sekunder.

#### a.. Data primer

Data yang diperoleh dari database sistem informasi akademik di Universitas Stikubank Semarang

#### b. Data Sekunder

Data yang diperoleh dengan membaca dan mempelajari referensi mengenai *instant messaging*, *protokol jabber*, *xml*, *chat bot*, *system question answering*.

### 3. Teknik Pengumpulan Data

Pengumpulan data mempunyai tujuan mendapatkan materi – materi yang mempunyai keterkaitan dengan topik penelitian. Pengumpulan data dimaksudkan agar mendapatkan bahan-bahan yang relevan, akurat dan reliable. Maka teknik pengumpulan data yang dilakukan dalam penelitian ini adalah dengan metode

Observasi, Studi Pustaka dan Metode pengembangan dengan menggunakan model prototyping.

## INSTANT MESSAGING

*Instant messaging* merupakan fasilitas komunikasi *chatting* untuk para pengguna internet. Dengan menggunakan fasilitas ini, *user* dapat berkomunikasi dengan cara mengirimkan pesan berupa *text* dengan *user* lain. Selain itu, Instan Messaging juga berfungsi untuk tukar menukar file secara *peer to peer*. Saat ini Instan Messaging yang terkenal adalah Yahoo Messenger dan MSN Messenger. Namun kita dapat membuat Instan Messenger sendiri menggunakan protokol Jabber serta aplikasi yang Open Source dan serta memodifikasi client sesuai dengan keinginan kita. Bahkan saat ini, Google mengembangkan Instan Messengernya, *Gtalk*, yang juga berbasis protokol XMPP (Jabber).

## Sekilas Tentang Protokol Jabber

Jabber adalah sebuah protokol XML yang terbuka untuk pertukaran *message* dan *presence* yang *real-time* antara dua *user* di dalam jaringan Jabber. Banyak kegunaan teknologi Jabber, pada awalnya teknologi Jabber bersifat *asynchronous*, *platform* IM yang dapat digunakan secara luas dan jaringan IM berdasarkan fungsinya hampir sama dengan sistem IM yang resmi seperti *AOL Instant Messaging* (AIM) dan *Yahoo Instant Messaging* (jabber.org,2003).

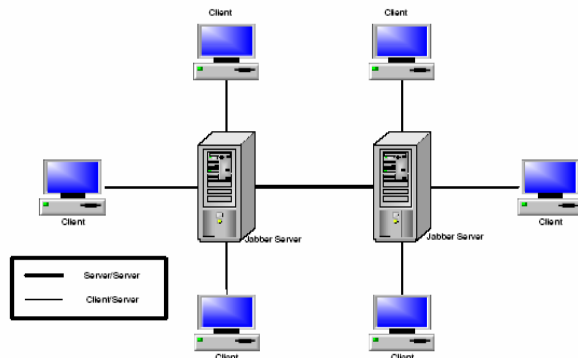
Sebagai usaha menjadikan Jabber sebagai protokol standar *Instant Messaging*, pada Juni 2000 komunitas Jabber telah mempublikasikan protokol tersebut sebagai *Request for Comment* (RFC) pada *Internet Engineering Task Force* (IETF) sebagai bagian dari standar *Instant Messaging and Presence Protocol* (IMPP), tetapi IMPP ini tidak berjalan sukses. Pada bulan Mei 2001, *Jabber Community* dan *Jabber Inc.* membuat *Jabber Software Foundation* untuk menyediakan asisten organisasi secara langsung (*direct organizational assistance*) dan asisten teknis secara tidak langsung terhadap komunitas Jabber.

Pada tahun 2002, *Internet Engineering Steering Group* (IESG) menyetujui formasi *Extensible Messaging and Presence Protocol Working Group* (XMPP) dengan *Internet Engineering Task Force* (IETF). Ruang lingkup utama *working group* adalah membuat *XML stream* termasuk pada level *security* dan autentikasi, elemen data dan *namespace* yang dibutuhkan untuk mencapai dasar IM dan *Presence*. (Cover,2002)

XMPP *working group* menerbitkan *XMPP Core Internet-Draft* sebagai dokumen yang menggambarkan fitur-fitur utama *Extensible Messaging* dan protokol *Presence*. Makalah XMPP ini memuat protokol Jabber yang bekerja pada sistem keamanan *client-server* dan *server-server*.

**Teknologi Protokol Jabber**

Jabber terkenal dengan arsitektur *client-server*nya, *client* Jabber dapat berkomunikasi dengan server Jabber pada *domain* Jabber mereka. *Domain* Jabber memiliki keuntungan yaitu kemampuannya dalam memisahkan zona komunikasi, yang ditangani oleh server Jabber yang berbeda, tidak seperti kebanyakan sistem IM lainnya yang menggunakan satu server terpusat untuk seluruh zona komunikasi (Shigeoka,2002). Gambar 1 menunjukkan *stream* Jabber *client-server*.

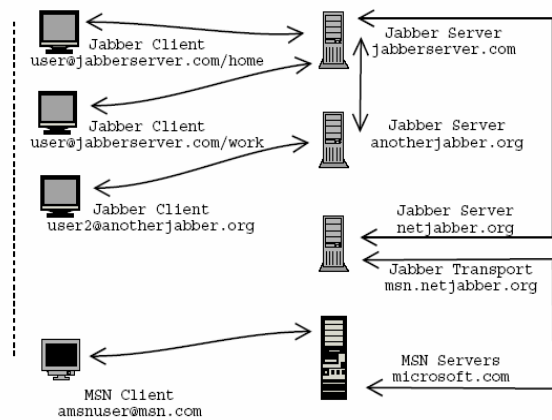


**Gambar 1.** Aliran *client-server* Jabber

XMPP merupakan protokol hasil formalisasi IETF dari *streaming* protokol standar XML, yang dikembangkan oleh *Jabber Community*. Protokol ini menghadirkan fitur lengkap untuk *Instant Messaging* dan *Presence* di atas *data transport layer* yang bersifat

*dedicated*. Protokol ini telah stabil sejak tahun 1999. Jabber/XMPP adalah sebuah protokol yang telah didokumentasikan dengan baik dari seluruh protokol yang ada dan mudah untuk dipahami. *working group* adalah untuk mengeksplorasi

Teknologi dasar dari XMPP menyangkut proses negosiasi XML *stream* antara *client* dan server, dengan menggunakan *Simple Autentication and Security Layer* (SASL) dan *Transport Layer Security* (TLS) untuk mengamankan pengiriman datanya. Setelah melakukan autentikasi, selanjutnya pengguna dapat mengirimkan fragmen-fragmen XML sebagai hasil dari menjalankan fungsi-fungsi IM, seperti mengirimkan pesan, chat dengan teman, merubah *status presence*, mengatur *contact list*, bergabung dengan *chatroom*, dan lain-lain. Server kemudian akan mengirimkan *message* kepada server lain melalui XML *stream* yang telah melalui proses negosiasi, berhubungan dengan syarat-syarat *security* untuk kemudian mencapai lokasi responden pengguna. XMPP kompatibel dengan teknologi Jabber yang sudah ada, sehingga menjamin interoperabilitas dengan jaringan yang ada saat ini. Aliran data pada protokol Jabber dapat dilihat pada Gambar 2



**Gambar 2.** Aliran data pada protokol Jabber

Cara Jabber/XMPP bekerja sering digambarkan seperti sebuah *router XML* artinya jika pesan dikirim dalam bentuk paket XML dan *route*-nya (pesan tersebut akan dikirim ke lokasi yang berdasar *content*-nya). Jabber di desain serupa dengan HTTP dan email karena protokol

ini relatif baru sampai saat ini Jabber memiliki sistem keamanan yang lebih baik.

Jabber merupakan sistem jaringan terdistribusi yang menggunakan konektivitas *Domain Name Service* (DNS), Jabber mempunyai sebuah fasilitas *dial-back* yang tidak sama dengan *email* untuk menempatkan alamat, artinya seseorang yang melakukan *spamming* pada sebuah server dengan jumlah data yang besar secara cepat. *Password* dapat disimpan dan di autentikasi dengan berbagai cara termasuk menggunakan PGP/SSL.

Saat ini tersedia banyak dokumentasi tentang komunikasi Jabber/XMPP dan protokol yang hanya sekali untuk didokumentasi secara keseluruhan. Jabber *support* terhadap sejumlah skema autentikasi dari algoritma *Hashing plaintext* dan standard SASL. Dengan menggunakan Jabber, komunikasi *client* ke *server* melalui SSL dan beberapa *client* menggunakan PGP berdasarkan *software* enkripsi. Sistem Jabber dapat juga terhubung ke sistem lainnya dengan sesuatu yang disebut *transport* yang berdasarkan *client emulation* dan dapat dijalankan pada *server* Jabber berdasarkan interoperabilitas antar protokol.

Ditinjau dari sistem keamanan, pada protokol Jabber terjadi *client bugs* semacam *buffer overflow* yang berpengaruh pada versi khusus dari aplikasi yang secara langsung tidak dipengaruhi oleh virus atau *hacker*.

### Arsitektur Jabber

Jabber menggunakan arsitektur *client-server*, bukan arsitektur langsung *peer-to-peer* seperti yang digunakan oleh sistem *messaging* lainnya. Akibatnya, seluruh data Jabber dikirim dari satu *client* ke *client* lainnya harus melewati minimal satu *server* Jabber. *Client* Jabber terhubung pada sebuah *server* Jabber pada TCP melalui port 5222. Koneksi ini selalu *on* untuk *session client* yang berjalan pada *server*, artinya *client* tidak dapat mengumpulkan pesan sebagai sebuah *email client*. Sebuah pesan diharapkan tersedia pada *client* dan dengan segera diharapkan *client messenger* sepanjang *client* masih terhubung. Server akan dapat menjajaki (*tracking*) apakah *client* masih *online* atau tidak, dan ketika *client*

dalam kondisi *off-line* akan menyimpan beberapa pesan yang telah dikirim kepada *client* untuk menyediakan kapan dia akan terhubung lagi.

Kekhasan yang dimiliki oleh protokol Jabber antara lain *modular server* dan *simple client* yang penjelasannya sebagai berikut :

#### 1. Modular server

Server Jabber memiliki tiga peranan utama yaitu :

- a. Menangani koneksi *client* dan berkomunikasi secara langsung dengan *client* Jabber
- b. Berkomunikasi dengan server Jabber yang lain
- c. Mengkoordinasikan beragam komponen *server* yang diasosiasikan dengan *server*

*Server* Jabber di desain modular, dengan paket kode internal yang khusus sehingga dapat menangani fungsionalitasnya seperti registrasi, autentikasi, *present*, *contact list*, penyimpanan pesan yang berstatus *off-line* dan sebagainya. Selain itu *server* Jabber dapat dikembangkan dengan komponen eksternal yang memungkinkan *administrator server* untuk mensuplemen *server* pusat dengan layanan tambahan semacam gerbang untuk sistem *messaging* lainnya.

#### 2. Simple client

Satu kriteria desain sistem Jabber bahwa ia harus memiliki kemampuan untuk mendukung *client* yang sederhana misalnya koneksi telnet pada *port* yang benar. Dalam hal ini tentu saja arsitektur Jabber memberikan sedikit batasan pada *client*.

*Task-task* pada *client* Jabber harus dapat mengenal dan melengkapinya :

- a. Komunikasi dengan *server* Jabber melalui soket TCP
- b. Melakukan *parsing* dan interpretasi XML dengan format yang baik melalui XML stream
- c. Memahami tipe data utama Jabber (*message*, *presence* dan *iq*)

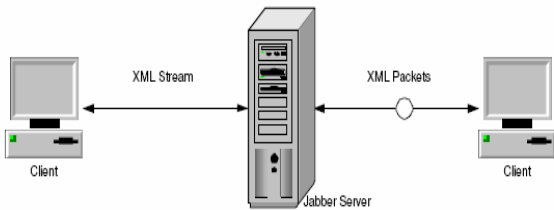
Keuntungan di dalam Jabber adalah dapat memindahkan kompleksitas dari *client* ke *server*. Secara praktis, banyak fungsi yang *low-level* pada *client* seperti proses *parsing XML* dan memahami tipe *data core* Jabber yang ditangani oleh *library-library client Jabber*, memungkinkan *client developer* untuk fokus pada *user interface* (Shigeoka,2002).

**FORMAT DATA XML**

Format data XML adalah bagian integral arsitektur Jabber karena sepenuhnya penting sehingga arsitektur secara fundamental dapat dikembangkan dan mampu diekspresikan dengan bentuk data yang terstruktur. Gambar 3 menunjukkan model *messaging Jabber* yang digabungkan dengan 4 elemen utama yaitu :

- a. Paket XML memuat data yang di *marked-up*,
- b. XML *stream* yang digunakan untuk transportasi paket XML ,

*Client* dan *server Jabber* yang dapat saling dipertukarkan



**Gambar 3.** Model *messaging Jabber*

Dengan terhubungnya *client* pada *server*, berarti membuka satu jalur ke *XML stream* dari *client* ke *server*, dan *server* merespon dengan satu jalur *XML stream* dari *server* ke *client*. Selanjutnya masing-masing *session* melibatkan dua *XML stream*. Seluruh komunikasi antara *client* dan *server* terjadi pada *stream* ini, contohnya adalah sebagai berikut :

```
<message
from='jolie@jabber.com/home'
to='aim@rhymbox.com/work'>

  <body>
    Hello, I
    need to
```

```
ask you
a
question
!?!<bod
y>
```

```
</message>
```

**Question Answering System (QA System)**

*QA system* atau *question-answering system* memberikan kemampuan pada sebuah mesin (komputer) untuk menginterpretasikan bahasa alami untuk melakukan dialog dengan pengguna hampir seperti dialog antara dua orang manusia dalam bahasa sehari-hari.

Dalam tingkatan bahasa yang dikemukakan oleh Noam Chomsky, *QA system* ini dapat digolongkan ke tingkatan bahasa tipe 0. Tingkatan bahasa itu sendiri terdiri dari empat buah tipe sebagaimana pada *hirarki Chomsky* pada tabel 1 berikut ini:

**Tabel 1 Hirarki Chomsky**

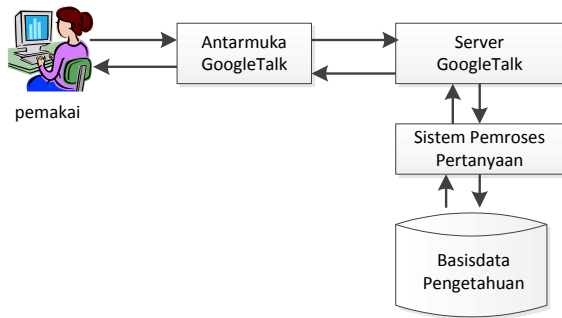
Bahasa	Mesin otomata	Batasan aturan Produksi
Regular / Tipe 3	Finite State Automata (FSA) meliputi Deterministic Finite Automata (DFA) dan Non-deterministic Finite Automata (NFA)	$\alpha$ adalah sebuah simbol variabel. B maksimal memiliki sebuah simbol variabel yang bila ada terletak di posisi paling kanan
Bebas Konteks / Context Free / Tipe 2	Push Down Automata (PDA)	$\alpha$ berupa sebuah simbol variabel
Context Sensitive / Tipe 1	Linear Bounded Automata	$ \alpha  \leq  \beta $
Unrestricted / Phase Structure / Natural Language / Tipe 0	Mesin Turing	Tidak ada batasan

Selanjutnya, aplikasi *QA System* saat ini lebih dikenal dengan nama *chat bot* atau *chatterbot* atau ada pula yang menyebutnya *chat robot*.

**HASIL DAN PEMBAHASAN**

**Arsitektur Sistem**

*QA system* merupakan sebuah sistem yang mengijinkan user menyatakan kebutuhan informasinya dalam bentuk yang lebih spesifik dan alami, yaitu dalam bentuk *natural language question*, dan tidak mengembalikan daftar dokumen yang harus disaring oleh user untuk menentukan apakah dokumen-dokumen tersebut mengandung jawaban atas pertanyaan, tetapi mengembalikan kutipan teks singkat atau bahkan frase sebagai jawaban.

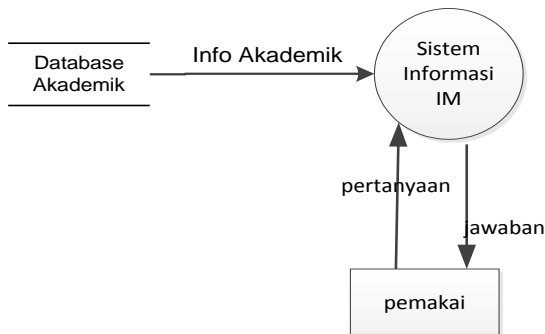


**Gambar 4.** Arsitektur Question Answering System

Gambar 4 menunjukkan arsitektur umum dari QA system yang tersusun atas enam tahapan proses, yaitu: *Piranti antar muka pemakai adalah Google Talk, Basisdata Pengetahuan adalah kumpulan masukan dari system QA, Pemroses Pertanyaan adalah actor utama Sistem QA yang memproses masukan dari pemakai, mencari jawaban, membuat keluaran dan mencatat ke log transaksi*

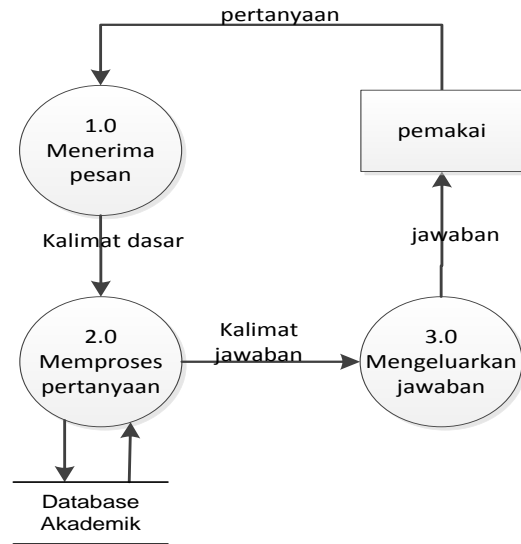
**Desain Sistem**

Pada gambar 5 dapat dilihat konteks diagram untuk desain sistem instant messaging akademik dalam penelitian ini. Ada dua eksternal entitas yaitu Sistem informasi akademik dan pemakai. Sistem informasi akademik merupakan sistem berisi data akademik. Sedangkan pemakai dapat meminta informasi tentang informasi akademik, krs, jadwal, dan nilai. Setelah system mendapat input dari pemakai, maka system selanjutnya menampilkan jawaban yang sesuai



**Gambar 5.** Konteks Diagram Sistem Instant Messing Informasi Akademik

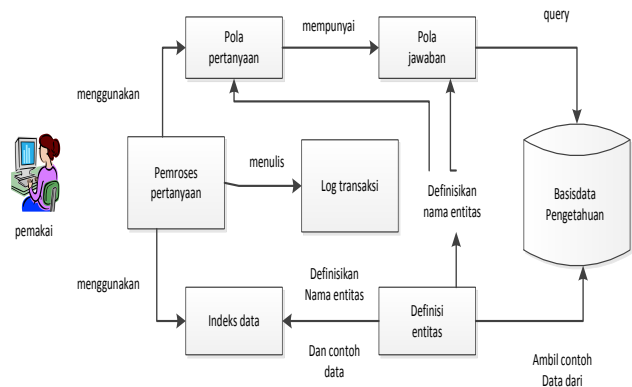
Sistem layanan informasi akademik instant messeging terdiri dari tiga bagian yaitu modul untuk menerima pesan, memproses pertanyaan dan mengeluarkan jawaban. Dapat dilihat pada data flow diagram rinci pada gambar 6



**Gambar 6.** Diagram Rinci

**Pemroses Pertanyaan**

Pemroses pertanyaan adalah media interaksi antara pemakai dan basisdata. Pemroses pertanyaan dapat diperlihatkan pada gambar 7



**Gambar 7.** Desain sistem secara logika

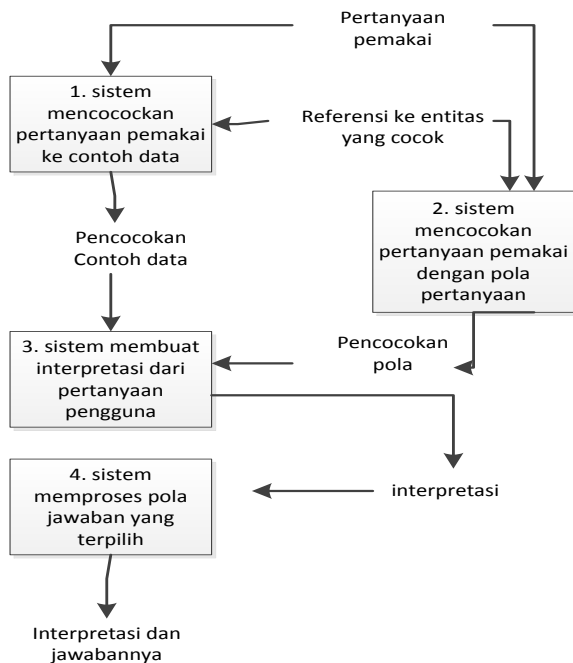
Komponen secara logika Pemroses Pertanyaan

- a. Definisi entitas memaparkan entitas yang diketahui oleh pemroses pertanyaan dan contoh data yang berkaitan dengan entitas

- tersebut.
- Indeks data adalah indeks dari contoh data yang dilingkupi definisi entitas. Indeks membantu secara cepat menemukan lokasi contoh data dan korespondensinya dengan entitas yang relevan dengan pertanyaan pemakai.
  - Sistem mencocokkan pertanyaan pemakai dengan sejumlah pola pertanyaan yang telah disimpan sebelumnya dan mengambil yang paling relevan.
  - Tiap pola pertanyaan mempunyai pola jawaban yang berasosiasi yang menghasilkan jawaban untuk pertanyaan pemakai. Komponen utama dari pola jawaban adalah pola query basisdata.
  - Pemroses pertanyaan terhubung dengan server talk.gmail.com
  - Pertanyaan pemakai akan disimpan di log transaksi

**Proses dan aliran data dari pemroses pertanyaan**

Proses utama dan aliran data dalam pemroses pertanyaan adalah sebagai berikut :



**Gambar 8.** Proses dan aliran data dari pemroses pertanyaan

Proses 1. Ketika seorang pemakai mengirimkan sebuah pertanyaan, system mengidentifikasi contoh data dan entitas yang berkorespondensi yang dinyatakan dalam pertanyaan.

Proses 2. Sistem mencocokkan pertanyaan pemakai ke pola pertanyaan dan mengambil satu pola pertanyaan yang secara semantic dekat dengan pertanyaan pemakai dan mereferensikan entitas yang sama dengan pertanyaan pemakai.

Proses 3. Untuk tiap pola pertanyaan yang diterima dari proses 2, sistem mengisi slot entitas dengan contoh data dari Proses 1 dan ambil sebuah interpretasi. Sistem memverifikasi apakah pertanyaan mempunyai jawaban.

Proses 4. Sistem memproses pola jawaban yang berasosiasi dengan interpretasi dan membuat jawaban sesuai dengan interpretasi.

**Pola Pertanyaan dan Model Konseptual dari Basisdata**

Pemodelan konsep adalah sebuah aktivitas yang mencakup pemunculan konsep, atribut-atributnya, hubungan dan pembatasan dari domain pengetahuan.

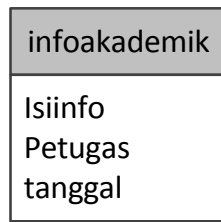
Selama konseptualisasi, informasi ditransformasikan menjadi kalimat, kalimat menjadi kalimat dasar dan kalimat dasar menjadi pasangan obyek-kaidah. Sebuah model konseptual memaparkan kalimat dasar mana yang masuk dan bu ke system informasi.

Jawaban pola dibuat dengan bantuan pola query basisdata=sebuah query basisdata formal yang mempunyai slot-slot entitas untuk contoh data. Setelah slot-slot diisi, pola menjadi query basisdata yang dapat dieksekusi secara biasa.

Berikut ini pola query yang berasosiasi dengan pola pertanyaan :

```
SELECT isiinfo FROM `infoakademik` limit 3
```

Kalimat bahasa alami dipetakan menjadi model ER



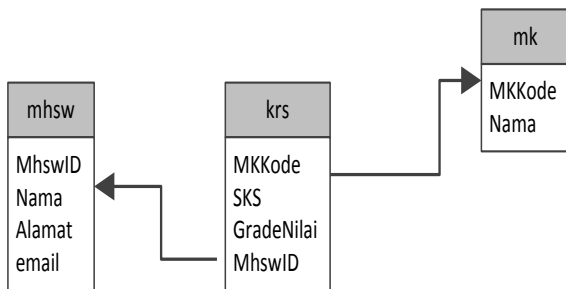
**Gambar 9.** Gambar Tabel Info Akademik

**Pertanyaan : krs**

Pola query :

```
SELECT krs.MKKode, krs.SKS, mk>Nama
FROM krs, mk WHERE krs.MKKode =
mk.MKKode AND krs.MhswID =
mhsW.MhswID and
mhsW.email='student.unisbank@gmail.com
```

Dipetakan ER



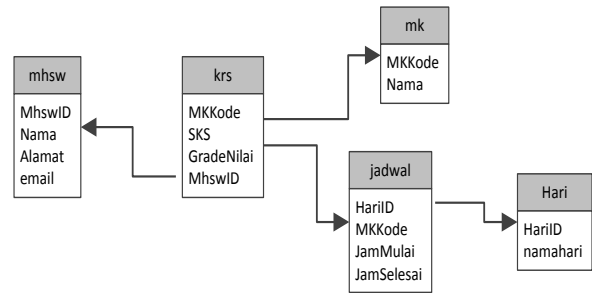
**Gambar 10.** Gambar Tabel KRS

**Pertanyaan : jadwal**

Pola query :

```
SELECT krs.MKKode, krs.SKS, mk>Nama,
Hari>Nama as namahari,jadwal.JamMulai,
jadwal.JamSelesai
FROM krs, mk, jadwal, Hari,mhsW WHERE
krs.MKKode = mk.MKKode AND
krs.MKKode = jadwal.MKKode AND
Hari.HariID = jadwal.HariID
AND krs.MhswID = mhsW.MhswID AND
mhsW.email='student.unisbank@gmail.com'
```

Dipetakan ER



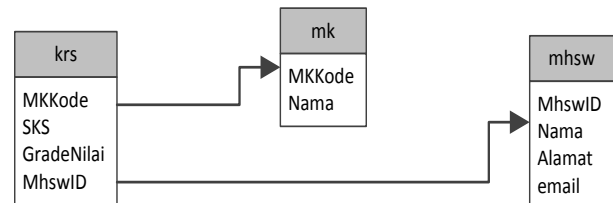
**Gambar 11.** Gambar Tabel Jadwal

**Pertanyaan : nilai**

Pola query :

```
SELECT krs.MKKode, krs.SKS,
mk>Nama,krs.GradeNilai
FROM krs, mk,mhsW WHERE krs.MKKode =
mk.MKKode AND krs.MhswID =
mhsW.MhswID and
mhsW.email='student.unisbank@gmail.com
```

Dipetakan ER

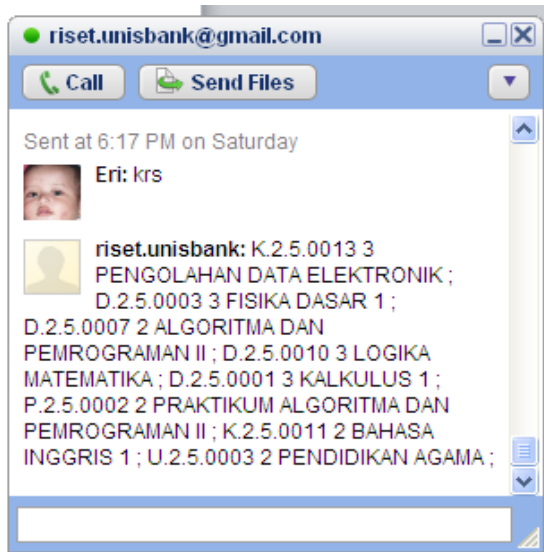


**Gambar 12.** Gambar Tabel Nilai

Gambar 5.1 menampilkan layar untuk berinteraksi dengan *agent* yang akan menjawab informasi layanan informasi akademik. Pertanyaan : Tampilkan KRS. Pertanyaan krs akan menampilkan krs untuk matakuliah yang diambil.



Maka jawaban dari chat bot :

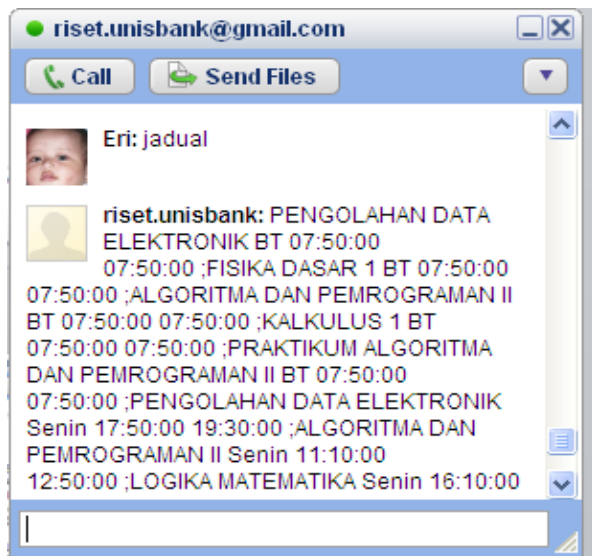


**Gambar 13.** Tampilan Pertanyaan KRS

**Pertanyaan 2: Tampilkan jadual**

Pertanyaan jadual akan menampilkan jadual untuk matakuliah yang diambil.

Maka jawaban dari chat bot :

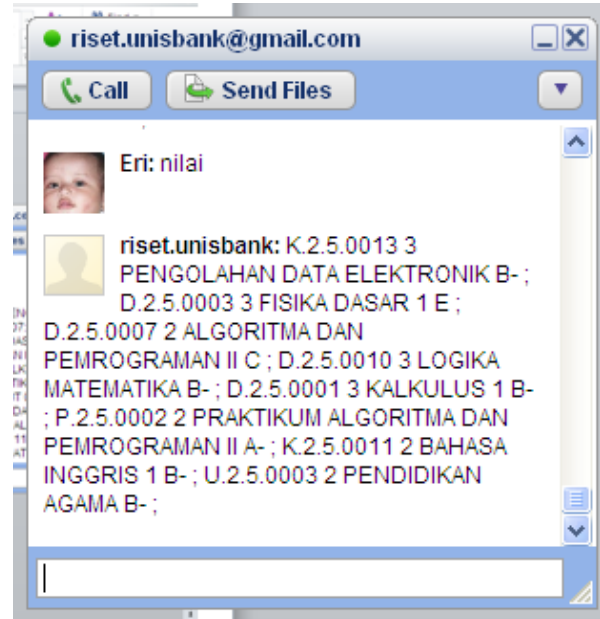


**Gambar 14.** Tampilan Pertanyaan jadual

**Pertanyaan : Tampilkan nilai**

Pertanyaan nilai akan menampilkan nilai untuk matakuliah yang diambil.

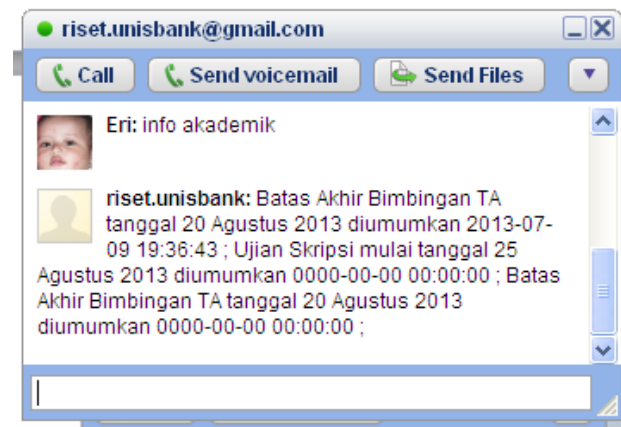
Maka jawaban dari chat bot :



**Gambar 15.** Tampilan Pertanyaan nilai

**Pertanyaan : info akademik**

Pertanyaan ini berkaitan dengan berita akademik. Informasi yang ditampilkan berupa pengumuman akademik yang terbaru.



**Gambar 16.** Tampilan Pertanyaan info akademik

**KESIMPULAN**

Dalam penelitian ini telah dibuat sistem layanan informasi akademik menggunakan Instant Messageng untuk mengelola layanan informasi akademik. Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan hal-hal sebagai berikut :

1. Sistem yang dibuat akan secara otomatis menjawab sesuai pertanyaan pemakai.
2. Pertanyaan dalam bahasa alami akan diubah menjadi perintah query oleh sistem. Selanjutnya sistem akan mengeksekusi query dan mengirimkan hasilnya ke pemakai.
3. Dengan memanfaatkan server Jabber talk.gmail.com, maka tidak diperlukan untuk membangun server Jabber sendiri.

Sistem telah diuji coba dan telah berjalan dengan baik

### **SARAN**

Dalam system ini masih terdapat kekurangan. Adapun kelemahan system ini antara lain adalah

1. Pertanyaan yang dilayani masih terbatas.
2. Antarmuka masih menggunakan Google Talk atau Psi.

### **DAFTAR PUSTAKA**

- Iain Shigeoka, (2002). *Instant Messaging in Java The Jabber Protocols*, Manning Publications Co.,
- Jabber.org FAQ. (n.d.). *jabber.org - the original XMPP instant messaging service*. <http://www.jabber.org/faq.html#jabber>. Diakses tanggal 12 Maret 2013.
- Miller J. P. Saint-Andre, (2003). “*XMPP Core Draft-IETF-XMPP-Core-12*” www page, May 2003, Expire on November 2,
- Robin Cover, (2002). “*IETF Charters Extensible Messaging and Presence Protocol(XMPP) Working Group.*,” www page, <http://xml.coverpages.org/>.
- Stephen Lee and Terence Smelser, (2002). *Jabber Programming*, Hungry Minds, Inc.,