

BAB I

PENDAHULUAN

1.1. Latar Belakang

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.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang dapat dirumuskan adalah bagaimana membangun Sistem Layanan Informasi Akademik menggunakan Instant Messenger.

Dalam penelitian ini ada beberapa pembatasan masalah yang dilakukan, yaitu:

1. Instant Messenger yang digunakan memanfaatkan layanan Google Talk.
2. Untuk dapat memanfaatkan fasilitas ini pemakai harus mempunyai account email gmail (xxxx@gmail.com).

BAB II

TUJUAN DAN MANFAAT PENELITIAN

2.1. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Sistem dapat menyebarkan informasi kepada seluruh mahasiswa.
2. Sistem dapat menjawab kebutuhan *request* informasi akademik dari civitas akademika..
3. Sistem mempunyai kemampuan untuk mengauthentikasi sms dari civitas akademika. Sehingga informasi hanya dapat di akses oleh civitas akademika yang berhak.

2.1. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah : hasil penelitian dapat digunakan untuk layanan informasi akademik.

BAB III

TELAAH PUSTAKA

3.1. 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 berbasiskan protokol XMPP (Jabber).

3.2 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*.

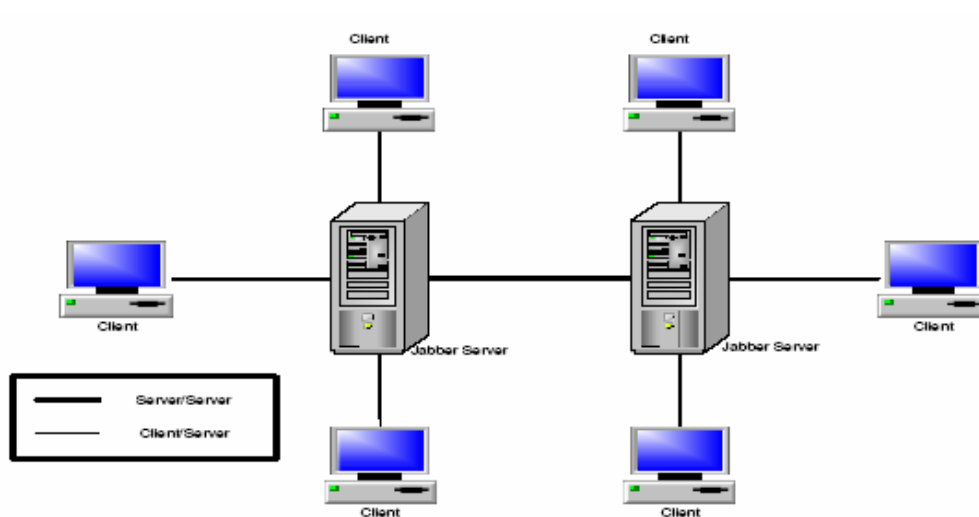
Pada tahun 2002, *Internet Engineering Steering Group* (IESG) menyetujui formasi *Extensible Messaging and Presence Protocol Working Group* (XMPP) dengan *Internet Engineering Task Force* (IETF). Ruang lingkup *working group* adalah untuk mengeksplorasi dan dimana protokol tersebut digunakan, memodifikasi protokol yang sudah ada agar dapat memenuhi RFC 2799 seperti persyaratan yang ditentukan dalam spesifikasi *Common Presence and Instant Messaging* (CPIM). Fokus utama *working group* adalah membuat *XML stream* termasuk *stream* 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*.

3.3. Tinjauan 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 5.1 menunjukkan *stream* Jabber *client-server*.

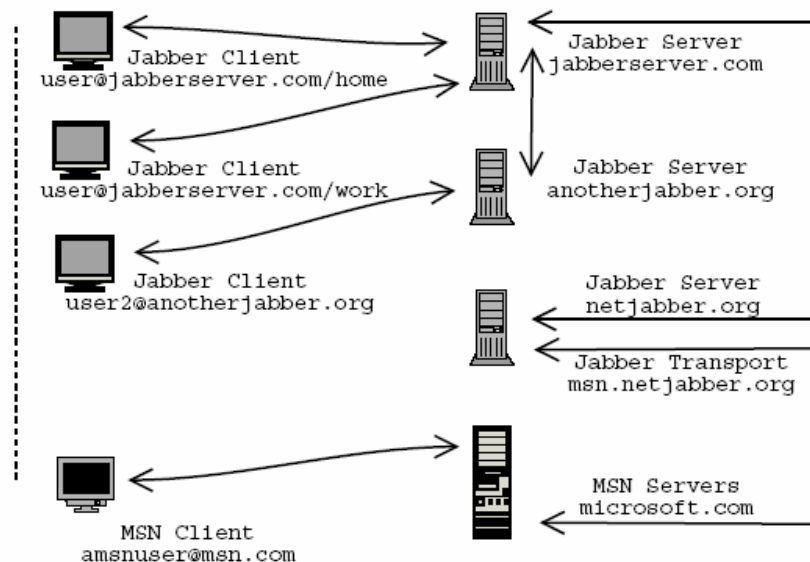


Gambar 3.1 Aliran *client-server* Jabber

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.

Teknologi dasar dari XMPP menyangkut proses negosiasi XML *stream*

antara *client* dan server, dengan menggunakan *Simple Authentication 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 3.2



Gambar 3.2 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*.

3.4. Arsitektur Jabber

3.4.1 Model *Client-Server*

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 :

- Menangani koneksi *client* dan berkomunikasi secara langsung dengan *client* Jabber
- Berkomunikasi dengan server Jabber yang lain
- 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 melengkapi :

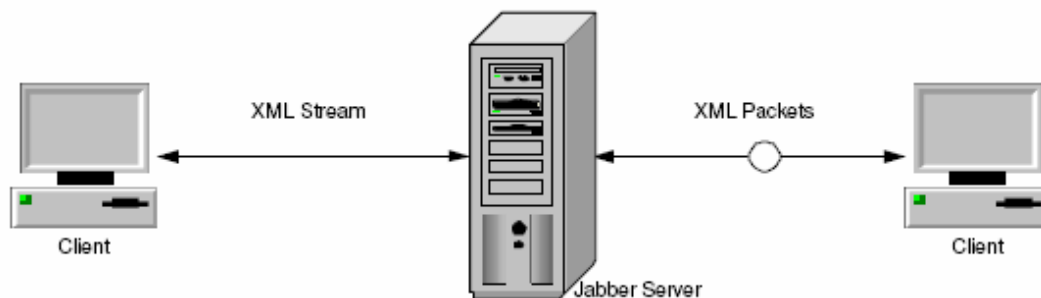
- Komunikasi dengan *server* Jabber melalui soket TCP
- Melakukan *parsing* dan interpretasi XML dengan format yang baik melalui XML stream
- Memahami tipe data utama Jabber (*message*, *presence* dan *iq*)

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).

3.4.2 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 5.3 menunjukkan model *messaging* Jabber yang digabungkan dengan 4 elemen utama yaitu :

- Paket XML memuat data yang di *marked-up*,
- XML *stream* yang digunakan untuk transportasi paket XML ,
- *Client* dan *server* Jabber yang dapat saling dipertukarkan.



Gambar 3.3 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
```

```
!<body>  
</message>
```

Ketika banyak *snippet* Jabber yang hanya sangat sederhana, format XML Jabber dapat juga ditingkatkan melalui *namespace* XML yang telah diatur oleh *Jabber Software Foundation* dan *namespace* disesuaikan untuk aplikasi yang khusus. Hal ini yang membuat Jabber menjadi *platform* yang *powerful* untuk memilih struktur data yang digunakan termasuk *XML Remote Procedure Calls* (XML-RPC), *Resource Description Framework Site Summary* (RSFSS) dan *Scalable Vector Graphics* (SVG) (jabber.org,2003).

3.4.3 Jaringan Terdistribusi

Jaringan terdistribusi dalam hal ini bagaimana sebuah *server* Jabber dapat berkomunikasi dengan *server* Jabber lainnya dan dapat diakses melalui internet. Masing- masing *user* terhubung pada *home server*, yang menerima informasi untuk mereka, selanjutnya *server* akan mentransfer data untuk kepemilikan *user*. Maka suatu domain dapat jalan pada *server* Jabber. Masing-masing fungsi *server* bebas terhadap yang lainnya, dan di-*maintain* sendiri di dalam daftar *user*. *User* khusus diasosiasikan dengan *server* yang khusus pula, dan dan alamat Jabber memiliki bentuk yang sama dengan alamat *email*. Jaringan yang terdistribusi ini menghasilkan sesuatu yang fleksibel, jaringan yang mampu terkontrol pada server yang memiliki skala yang lebih tinggi dibandingkan monolitik, tetapi dengan syarat bahwa layanan yang terpusat hanya dapat jalan pada vendor IM yang resmi (jabber.org,2003).

3.4.4 Standar Berdasarkan Pengalaman

Ada beberapa perbedaan entitas pada Jabber untuk dapat berkomunikasi dengan yang lainnya. Entitas ini dapat direpresentasikan berupa *transport*, *groupchat room* atau *single Jabber user*. *Jabber ID* telah digunakan secara internal dan eksternal untuk menyatakan kepemilikan atau *routing* informasi. Masing-masing *Jabber ID* memuat sekelompok order elemen. *JID* dibentuk dalam suatu *domain*, *node* dan *resource* dalam format [node@domain[/resource]] (Lee,2002), contohnya : riset.unisbank@jabber.com/home

3.5. Komponen Utama Protokol Jabber

Ada tiga komponen utama pada protokol Jabber yang diandalkan dengan mekanisme *messaging* :

1. Message

Protokol *message* pada kenyataannya adalah protokol yang paling sederhana dalam Jabber. Banyak *traffic* di dalam jaringan Jabber yang termasuk dalam protokol *message*. Message terdiri dari 4 atribut, dan *zero* atau beberapa *child element*. Attribute dalam *message* adalah :

- to** : jenis yang diharapkan oleh pihak penerima pesan
- from** : jenis pesan yang dikirim
- id** : sebuah identifier unik yang bersifat opsional dengan tujuan dapat menjejaki *message*
- type** : sebuah spesifikasi opsional dari konteks percakapan sebuah *message*

Sedangkan atribut pada elemen anak antara lain ;

body : isi tekstual dari *message*, secara normal termasuk tetapi tidak dibutuhkan.

subject : subjek dari *message*

thread : string acak yang di-*generated* oleh pengirim, digunakan untuk *tracking* sebuah *thread conversation*.

error : deskripsi pesan kesalahan

Versi protokol Jabber XMPP saat ini menggunakan standar yang merepresentasikan seluruh atribut dan elemen anak yang ada pada *message protocol*. Contoh paket *message* ditunjukkan seperti berikut :

```
<message to='romeo@montague.net'  
from='Juliet@capulet.com/balcony' type='chat'>  
  <subject xml:lang='en'> Greeting !  
</subject>  
  <body xml:lang='en'> Hello!!  
</body>  
<thread>e0f92794b9683a38</thread>  
</message>
```

2. Presence

Protokol ini bertanggung jawab terhadap *subscription*, persetujuan, dan *update* informasi *presence* dalam komunitas Jabber. Atribut diasosiasikan dengan dengan protokol ini sama seperti pada *protocol message*, *presence* memiliki memiliki tipe atribut yang memiliki 7 *state* sebagai berikut ;

1. **unavailable** : *client* tidak lama tersedia untuk berkomunikasi
2. **subscribe** : pengirim mengirimkan *request* untuk *subscribe* terhadap *presence* penerima
3. **subscribed** : pengirim yang telah diizinkan terhadap recipient untuk menerima *presence* mereka.
4. **unsubscribe** : *subscription request* yang telah ditolak atau *subscription* yang telah di *cancel* sebelumnya.
5. **probe** : *request* dari *client* yang *presence* saat ini
6. **error** : pesan kesalahan yang berlangsung berdasarkan pemrosesan atau menyediakan paket *presence* yang telah dikirim sebelumnya.

Paket *presence* akan bernilai 0 atau 1 untuk masing-masing elemen anak berikut :

- show** : menggambarkan status yang tersedia dari entities atau *resource* yang spesifik. show memiliki empat nilai yang digunakan :
- away : *temporarily away*
 - chat : bebas untuk *chat*
 - xa : *extended away*
 - dnd : *do not disturb*
- status** : merupakan deskripsi bahasa natural yang bersifat opsional yang mendeskripsikan status yang tersedia.
- priority** : bilangan *integer* bukan negatif yang menampilkan level

prioritas pada

resource yang terkoneksi, dengan 0 sebagai prioritas terendah.

error : deskripsi pesan kesalahan

Contoh paket *presence* adalah sebagai berikut :

```
<presence> from='juliet@capulet.com/balcony'  
    to='romeo@montague.net/orchard'>  
<show> away </show>  
<status> be right back </status>  
<priority> 0 </priority>  
</presence>
```

Paket *presence* dapat memuat *namespace* yang sesuai dengan elemen anak yang tidak mengganggu struktur *namespace* dan tag yang tersedia.

3. Info/Query

Protokol IQ adalah protokol Jabber yang terakhir dan yang paling peduli dibandingkan *message* dan protokol *presence*. IQ adalah protokol *request-response* yang umum sehingga di desain secara mudah untuk dikembangkan seperti HTTP yang merupakan medium *request-respon*. *Content* data dari *request* dan *respon* yang ditentukan dengan deklarasi *namespace* elemen anak secara langsung dari elemen IQ. Atribut diasosiasikan dengan protokol IQ memiliki atribut yang sama dengan protokol *message* dan *presence* kecuali jika protokol tersebut memiliki tipe atribut yang berbeda. Tipe atribut pada protokol *info/query* memiliki 4 nilai yang dapat digunakan ;

1. **get** : informasi *request*
2. **set** : menyediakan data yang dibutuhkan
3. **result** : respon terhadap *get* dan *set request* yang sukses
4. **error** : kesalahan yang terjadi dalam pemrosesan dan layanan *get* dan *set request*

Berikut adalah contoh paket IQ yang memblok *incoming message* dari JID yang khusus:

```

<iq type='set' id='msg1'>
  <query xmlns='jabber:iq:privacy'>
    <list name='message-jid-example'>
      <item type='jid' value='tybalt@capulet.com'
        action='deny'
        order='3'>
        <message/>
      </item>
    </list>
  </query>
</iq>

```

Protokol IQ ini sangat penting jika ingin membangun *server* berdasarkan kebijakan keamanan sistem yang harus dipenuhi oleh *client*. Jika sistem keamanan *client* telah terpenuhi maka harus mendukung pula terhadap sistem keamanan pada sisi *server*.

3.6. 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* berikut ini:

Tabel 3.1 Hirarki Chomsky

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

3.7. Chat Bot

Kata *bot* diambil dari kata *robot*, yang dapat diartikan sebagai sebuah program komputer yang berfungsi mengumpulkan informasi atau memberikan layanan yang biasanya telah dijadwalkan sebelumnya. Sebuah *bot* (atau biasa juga dikatakan sebagai sebuah *agent*) biasanya mencari informasi di internet,

mengumpulkan informasi yang sesuai dengan yang kita inginkan, dan memberikannya pada kita secara periodik atau sesuai dengan perintah kita.

Bot biasa digunakan dalam IRC (*Internet Chat Groups*) di mana *bot* tersebut melakukan berbagai fungsi seperti menyapa partisipan baru dan memonitor penggunaan kata-kata yang tidak pantas. Sekarang ini, *bot* dapat mengumpulkan informasi, mengolah informasi tersebut, dan memberikan respon terhadap pengguna dan *event* yang terjadi di internet. Beberapa jenis *bot* bekerja secara *silent* di internet tanpa sepengetahuan kita namun ada pula *bot* yang bekerja secara interaktif yang dapat kita ajak berbicara yaitu *chat bot*.

Chat bot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunanya dalam bahasa alami. Karena *chat bot* hanya sebuah program, dan bukan robot (*chat bot* tidak memiliki tubuh dan tidak memiliki mulut sehingga tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antara manusia sebagai pengguna dengan *chat bot* dilakukan dengan cara mengetik apa yang akan dibicarakan dan *chat bot* akan memberikan respon.

BAB IV

METODOLOGI 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. Teknik Pengumpulan Data

Pengumpulan data dimaksudkan agar mendapatkan bahan-bahan yang relevan, akurat dan reliable. Maka teknik pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut :

a. Observasi

Dengan melakukan pengamatan dan pencatatan secara sistematis tentang hal-hal yang berhubungan dengan basis data dokumen teks dan kemampuan pencarian kemiripan dokumen.

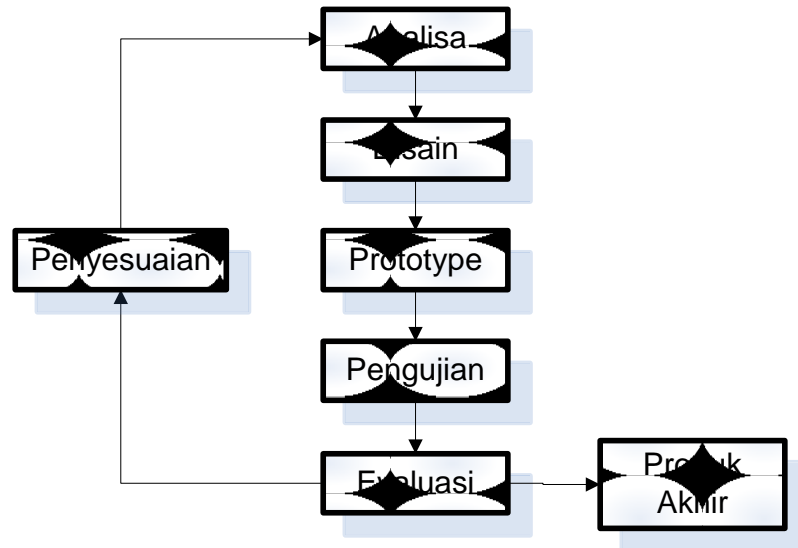
b. Studi Pustaka

Dengan pengumpulan data dari bahan-bahan referensi, arsip, dan dokumen yang berhubungan dengan permasalahan dalam penelitian ini.

3. Metode Pengembangan

Penelitian ini menggunakan model *prototyping*. Di dalam model ini sistem dirancang dan dibangun secara bertahap dan untuk setiap tahap pengembangan dilakukan percobaan-percobaan untuk melihat apakah sistem

sudah bekerja sesuai dengan yang diinginkan. Sistematika model *prototyping* terdapat pada gambar 4.1 memperlihatkan tahapan pada prototyping.



Gambar 4.1 Tahapan Prototyping (Pressman, 2001)

Berikut adalah tahapan yang dilakukan pada penelitian ini dengan metode pengembangan prototyping

a. **Analisa**

Pada tahap ini dilakukan analisa tentang masalah penelitian dan menentukan pemecahan masalah yang tepat untuk menyelesaikannya.

b. **Disain**

Pada tahap ini dibangun rancangan sistem dengan beberapa diagram bantu seperti Data Flow Diagram.

c. **Prototype**

Pada tahap ini dibangun aplikasi berbasis web yang sesuai dengan disain dan kebutuhan sistem.

d. Pengujian

Pada tahap ini dilakukan pengujian pada pustaka fungsi yang sudah dibangun.

e. Evaluasi

Pada tahap ini dilakukan evaluasi apakah performa aplikasi sudah sesuai dengan yang diharapkan, apabila belum maka dilakukan penyesuaian-penyesuaian secukupnya.

f. Penyesuaian

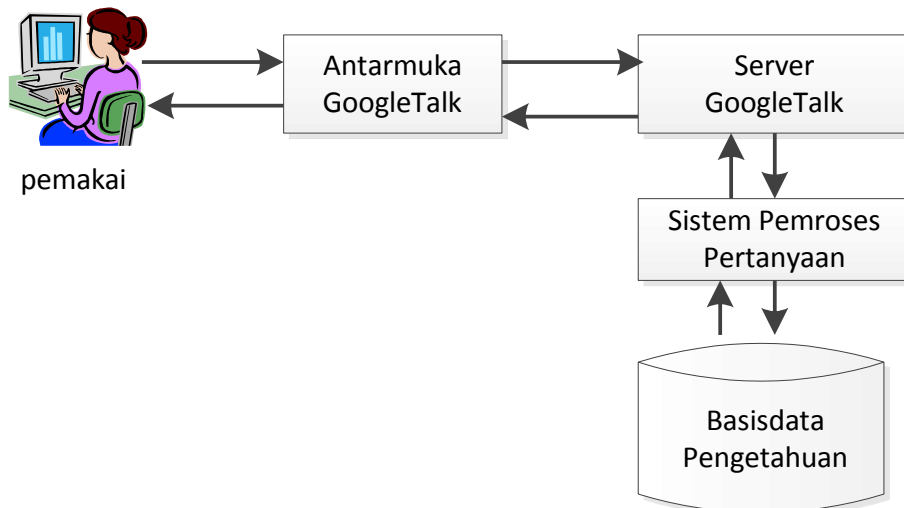
Tahap ini dilakukan apabila pada evaluasi performa aplikasi kurang memadai dan dibutuhkan perbaikan, tahap ini melakukan penyesuaian dan perbaikan pada aplikasi sesuai dengan kebutuhan

BAB V

HASIL DAN PEMBAHASAN

5.1. Arsitektur Sistem

QA system merupakan sebuah sistem yang memungkinkan 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.



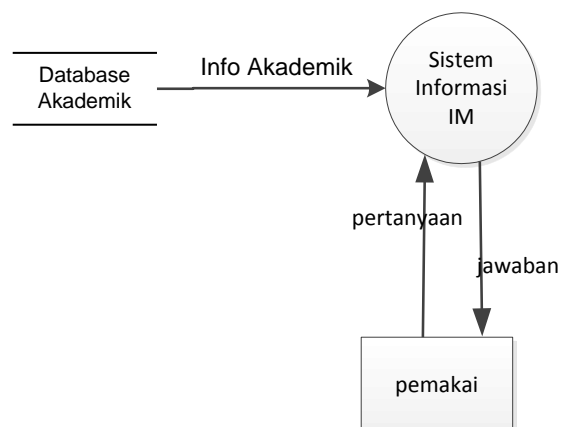
Gambar 5.1. Arsitektur Sistem

Gambar 5.1 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

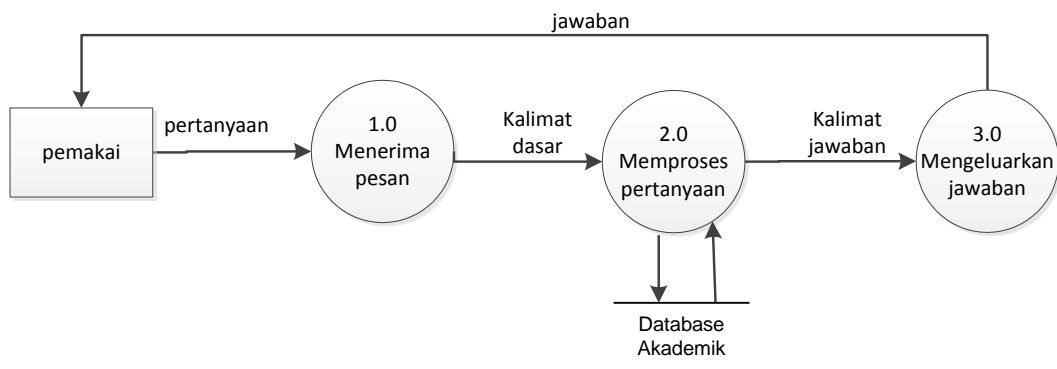
5.2. Desain Sistem

Pada gambar 5.2 dapat dilihat konteks diagram untuk desain sistem instant messaging akademik dalam penelitian ini. Ada satu eksternal entity yaitu pemakai. Sistem akan menggunakan database akademik yang sudah ada. Eksternal entity 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 5.2. Diagram konteks sistem

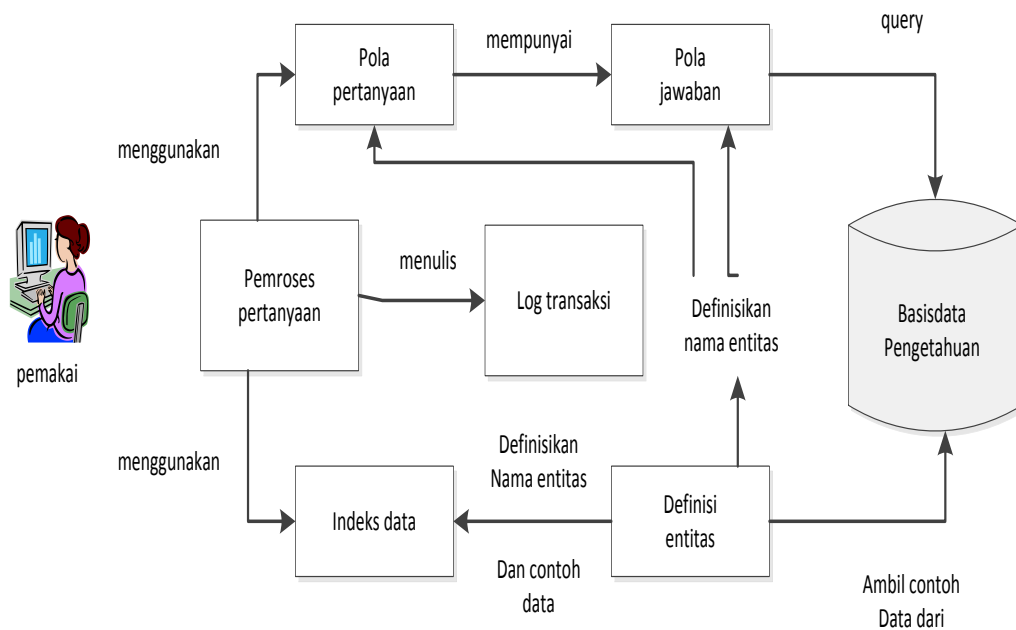
Sistem layanan informasi akademik instant messaging terdiri dari tiga bagian yaitu modul untuk menerima pesan, memproses pertanyaan dan mengeluarkan jawaban. Dapat dilihat pada data flow diagram rinci pada gambar 5.3



Gambar 5.3. Diagram rinci

5.3. Pemroses Pertanyaan

Pemroses pertanyaan adalah media interaksi antara pemakai dan basisdata.



Gambar 5.4 Desain sistem secara logika

Komponen secara logika Pemroses Pertanyaan

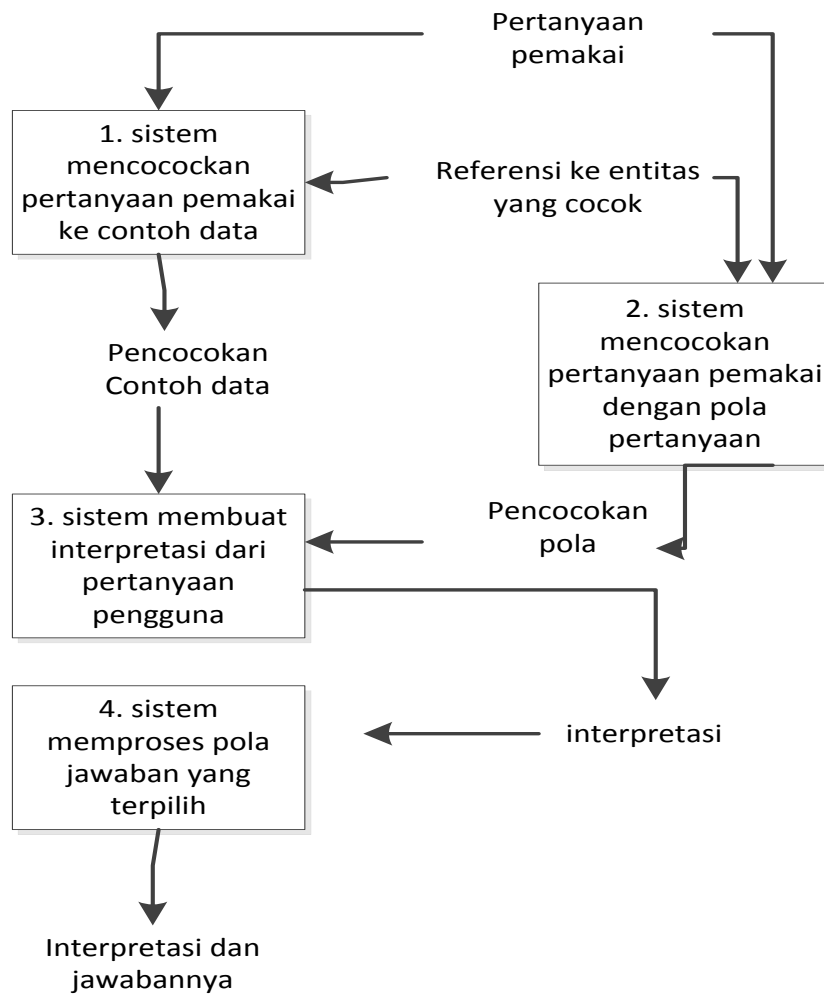
- Definisi entitas memaparkan entitas yang diketahui oleh pemroses pertanyaan dan contoh data yang berkaitan dengan entitas tersebut.
- Indeks data adalah indeks dari contoh data yang dilingkupi definisi

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

5.4. Proses dan aliran data dari pemroses pertanyaan

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



Gambar 5.5 Proses dan aliran data dari pemroses pertanyaan

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

5.5. 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

infoakademik
Isiinfo Petugas tanggal

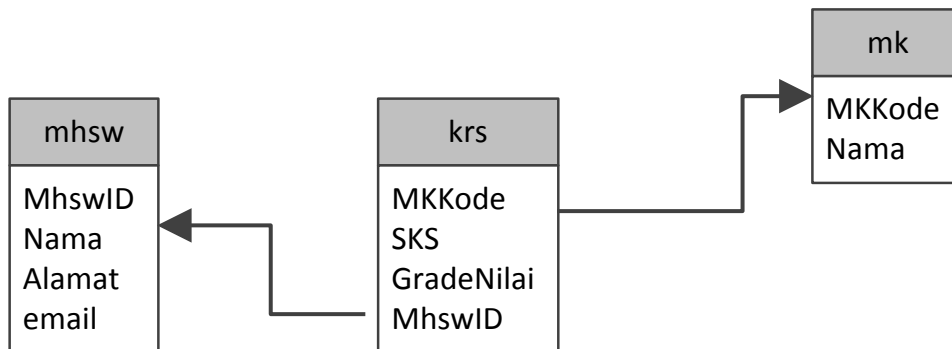
Gambar 5.6. Gambar Tabel Info Akademik

Pertanyaan : krs

Pola query :

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

Dipetakan ER



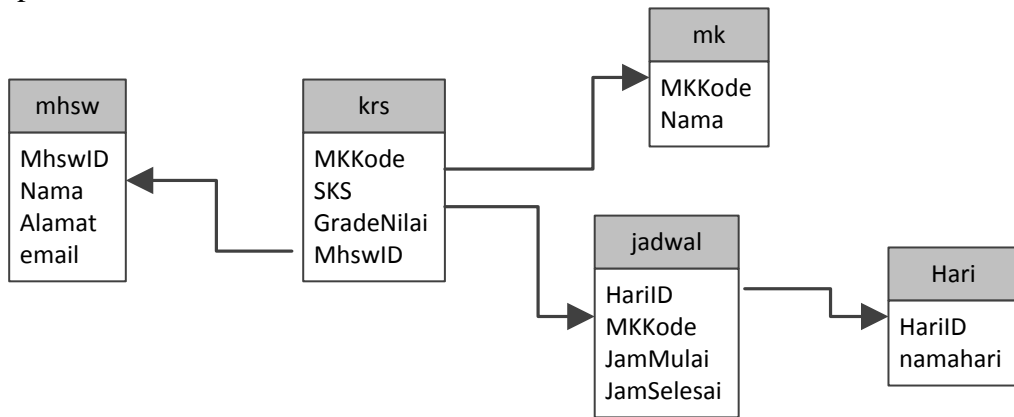
Gambar 5.7. Gambar Tabel KRS

Pertanyaan : jadual

Pola query :

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

Dipetakan ER



Gambar 5.8. Gambar Tabel Jadwal

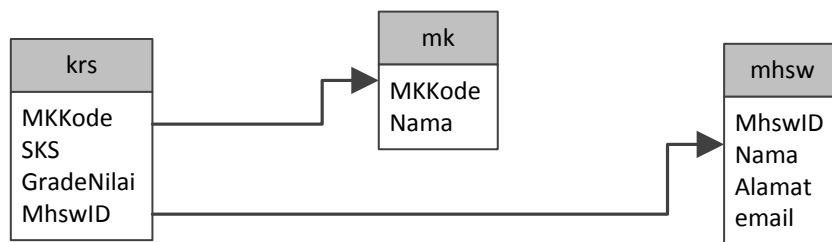
Pertanyaan : nilai

Pola query :

```

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

Dipetakan ER

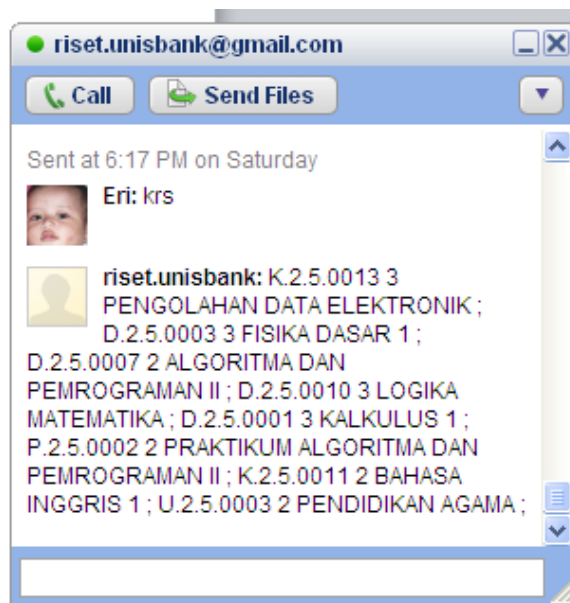


Gambar 5.9. Gambar Tabel Nilai

5.3. Tampilan

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 :

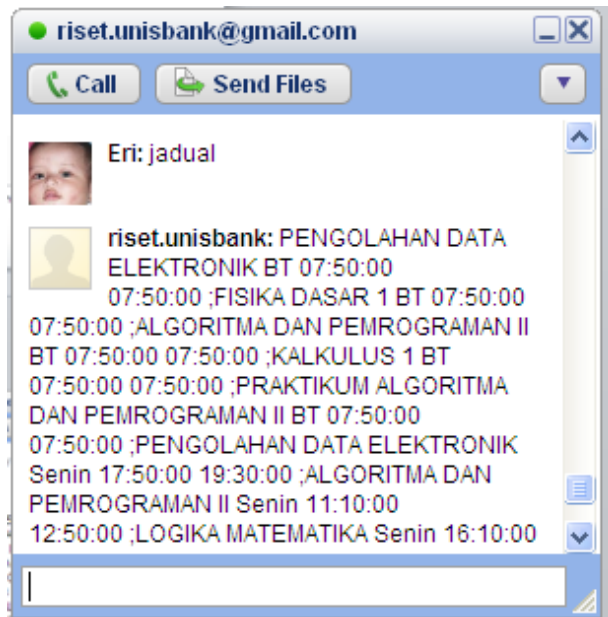


Gambar 5.10. Tampilan Pertanyaan KRS

Pertanyaan 2: Tampilkan jadwal

Pertanyaan jadwal akan menampilkan jadwal untuk matakuliah yang diambil.

Maka jawaban dari chat bot :

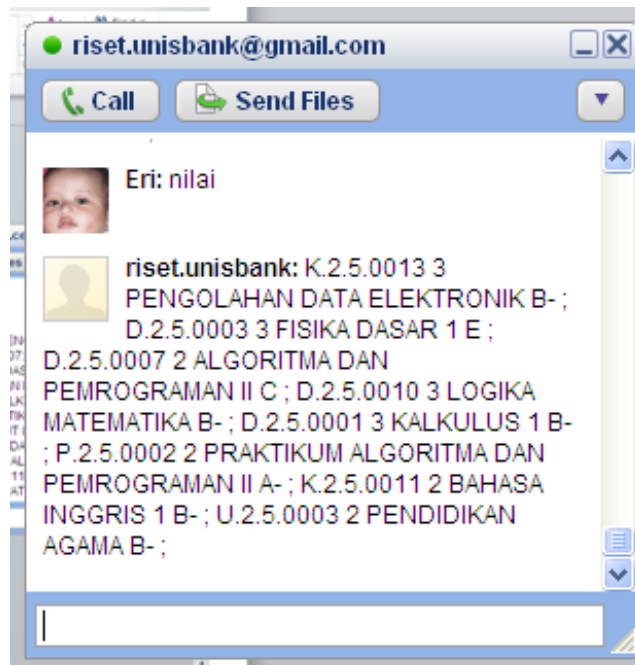


Gambar 5.11. Tampilan Pertanyaan jadwal

Pertanyaan : Tampilkan nilai

Pertanyaan nilai akan menampilkan nilai untuk matakuliah yang diambil.

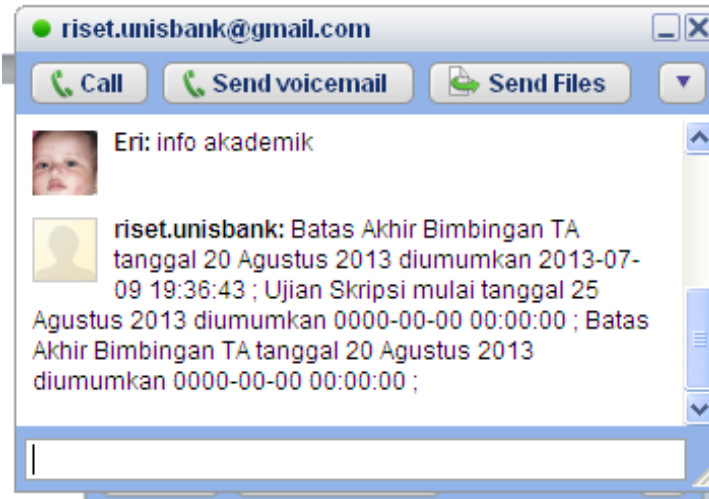
Maka jawaban dari chat bot :



Gambar 5.12. Tampilan Pertanyaan nilai

Pertanyaan : info akademik

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



Gambar 5.13. Tampilan Pertanyaan info akademik

BAB VI

KESIMPULAN DAN SARAN

6.1. 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

6.2. Saran

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

- a. Pertanyaan yang dilayani masih terbatas.
- b. Antarmuka masih menggunakan Google Talk atau Psi.

Pada penelitian selanjutnya akan dibangun :

1. Sistem yang mampu mengolah pertanyaan menggunakan kalimat bahasa Indonesia yang lebih kompleks.
2. Sistem antarmuka yang berjalan di gadget yang menggunakan Android.

DAFTAR PUSTAKA

Iain Shigeoka, 2002, *Instant Messaging in Java The Jabber Protocols*, Manning Publications Co.,

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.,

<http://www.jabber.org> “*What Is jabber,*” www page, 2003,.

DAFTAR RIWAYAT PENELITI (KETUA)

I. DATA DIRI

1. Nama Lengkap : Eri Zuliarso, Drs., M.Kom
2. NIY : YU.2.10.11.097
3. Golongan/Pangkat : IIB / Penata Muda Tk II
4. Jabatan Fungsional : Asisten Ahli
5. Tempat, Tgl. Lahir : Temanggung, 23 November 1968
6. Jenis Kelamin : Laki-laki
7. Alamat Rumah : Jl. Pucang Permai III/2, Mranggen, Demak
8. Telp / Faks / e-mail : 085876470885 / / ezuliarso@yahoo.com
9. Alamat Kampus : Jl. Trilomba Juang 1 Semarang
10. Telp / Faks / e-mail : 8311668 / 8443240 / info@unisbank.ac.id

II. RIWAYAT PENELITIAN

No	Judul	Tahun	Keterangan
1	Membangun Perpustakaan Digital Jurnal Elektronik	2013	Anggota
2	Membangun Perpustakaan Digital Buku Elektronik	2012	Ketua
3	Klasifikasi Berita Menggunakan Ontologi	2012	Anggota
4	Klastering Dokumen Berita dari Web Menggunakan Algoritma Single Pass Clustering	2011	Anggota
5	Prototipe Mesin Pencari Dokumen Teks	2010	Anggota

6	Aplikasi Web Crawler berdasarkan ontologi	2009	Anggota
7	Aplikasi Web Crawler berdasarkan breadth first search dan back link	2009	Anggota

Demikian daftar riwayat penelitian ini dibuat dengan sebenarnya.

Semarang, 17 Juli 2013

Drs. Eri Zuliarso, M.Kom

DAFTAR RIWAYAT PENELITI (ANGGOTA)

I. DATA DIRI

1. Nama Lengkap : Herny Februariyanti, ST., M.Cs
2. NIY : YS.2.01.01.035
3. Golongan/Pangkat : III C / Penata Muda
4. Jabatan Fungsional : Lektor
5. Tempat, Tgl. Lahir : Blora / 14 Pebruari 1973
6. Jenis Kelamin : Perempuan
7. Alamat Rumah : Kendeng V / 12 Semarang
8. Telp / Faks / e-mail : 08156545909/./herny@unisbank.ac.id
9. Alamat Kampus : Jl. Trilomba Juang 1 Semarang
10. Telp / Faks / e-mail : 8311668 / 8443240 / info@unisbank.ac.id

II. RIWAYAT PENELITIAN

No	Judul	Tahun	Keterangan
1	Membangun Perpustakaan Digital Jurnal Elektronik	2013	Ketua
2	Membangun Perpustakaan Digital Buku Elektronik	2012	Anggota
3	Klasifikasi Berita Menggunakan Ontologi	2012	Ketua
4	Klastering Dokumen Berita dari Web Menggunakan Algoritma Single Pass Clustering	2011	Ketua
5	Aplikasi Generator Konten untuk Meningkatkan Peringkat Situs pada Halaman Hasil Mesin Pencari.	2010	Anggota
6	Prototipe Mesin Pencari Dokumen Teks	2010	Ketua
7	Aplikasi Pengindeks Kata Berbasis Web Pada Dokumen	2009	Ketua

	Teks Berbahasa Indonesia Untuk Keperluan Temu Kembali Informasi.		
8	Hierarchical Agglomerative Clustering untuk Sistem Temu Kembali Dokumen Bahasa Indonesia	2009	Ketua
9	Pengindeks Kata Dokumen Teks dengan Menggunakan Aplikasi Berbasis Web	2009	Ketua

Demikian daftar riwayat penelitian ini dibuat dengan sebenarnya.

Semarang, 17 Juli 2013

Herny Februriyanti, ST., M.Cs

DAFTAR RIWAYAT PENELITI (ANGGOTA)

1. Nama Lengkap : Setyawan Wibisono, S.Kom., M.Cs
2. NIY/NIP : 19730607 200501 1 001
3. Golongan/Pangkat : IIIB / Penata Muda Tk. I
4. Jabatan Fungsional : Lektor
5. Tempat, Tgl. Lahir : Ambarawa, 6 Juli 1973
6. Jenis Kelamin : Laki-Laki
7. Alamat Rumah : Jl. Sinar Gemah Timur No. 964 H Sinar
Waluyo
8. Telp / Faks / e-mail : 081329979793 / .. / elestari@unisbank.ac.id
9. Alamat Kampus : Jl. Trilomba Juang 1 Semarang
10. Telp / Faks / e-mail : 8311668 / 8443240 / info@unisbank.ac.id

II. RIWAYAT PENELITIAN

No.	Judul	Tahun	Keterangan
1	Akses Node Dokumen XML Menggunakan XPath untuk Perolehan Data pada Basisdata Perundangan Pemerintah Kota Semarang	2012	Anggota
2	Implementasi XQuery Dalam Basisdata XML Pada Sistem Informasi Perundangan Pemerintah Provinsi Jawa Tengah	2012	Ketua
3	Query Bahasa Indonesia Pada Basis Data Soal Ujian Di Fakultas Teknologi Informasi Universitas Stikubank Dengan Format Data XML	2011	Anggota

4	Aplikasi Generator Nilai Tag Meta (Meta Tag Value) Pada Halaman Web Untuk Meningkatkan Peringkat Temu Kembali Informasi Mesin Pencari	2011	Anggota
5	Aplikasi Pengelolaan Peraturan Daerah Provinsi Jawa Tengah Menggunakan Basis Data XML	2010	Ketua
6	Adaptive Tutoring System Untuk Pembelajaran Bahasa Pemrograman Di Universitas Stikubank Semarang	2010	Anggota
7	Aplikasi Pengindeks Kata Berbasis Web Pada Dokumen Teks Berbahasa Indonesia Untuk Keperluan Temu Kembali Informasi	2009	Anggota
8	Implementasi Stemmer Bahasa Indonesia Tanpa Kamus Berbasis Web Untuk Keperluan Temu Kembali Informasi	2009	Ketua

Demikian daftar riwayat penelitian ini dibuat dengan sebenarnya.

Semarang, 17 Juli 2013

Setyawan Wibisono, S.Kom., M.Cs

DAFTAR RIWAYAT PENELITI (MAHASISWA)

I. DATA DIRI

1. Nama Lengkap : Tri Juwitaningrum
2. NIM : 09.01.53.0195
3. Golongan/Pangkat : -
4. Jabatan Fungsional : -
5. Tempat, Tgl. Lahir : Blora, 27/05/1990
6. Jenis Kelamin : Perempuan
7. Alamat Rumah : Jln. Karonsih Timur IX 337 Semarang
8. Telp / Faks / e-mail : 081389666091/
ningrumjuwita@yahoo.co.id
9. Alamat Kampus : Jl. Trilomba Juang 1 Semarang
10. Telp / Faks / e-mail : 8311668 / 8443240 / info@unisbank.ac.id

II. RIWAYAT PENELITIAN

No.	Judul	Tahun	Keterangan
1			

Demikian daftar riwayat penelitian ini dibuat dengan sebenarnya.
Semarang, 17 Juli 2013

Tri Juwitaningrum

DAFTAR RIWAYAT PENELITI (MAHASISWA)

I. DATA DIRI

1. Nama Lengkap : Rizka Nur Ariani
2. NIM : 09.01.53.0198
3. Golongan/Pangkat : -
4. Jabatan Fungsional : -
5. Tempat, Tgl. Lahir : Semarang, 12/08/1990
6. Jenis Kelamin : Perempuan
7. Alamat Rumah : Tlogosari
8. Telp / Faks / e-mail : 085641243963/rizka_a108@yahoo.co.id
9. Alamat Kampus : Jl. Trilomba Juang 1 Semarang
10. Telp / Faks / e-mail : 8311668 / 8443240 / info@unisbank.ac.id

II. RIWAYAT PENELITIAN

No.	Judul	Tahun	Keterangan

Demikian daftar riwayat penelitian ini dibuat dengan sebenarnya.

Semarang, 17 Juli 2013

Rizka Nur Ariani

LOKASI PENELITIAN

Peta kampus mugas - Unisbank



disinilah lokasi kampus mugas
Universitas Stikubank (UNISBANK)
Jalan Tri Lomba Juang No. 1 Semarang.

masih bingung mencari? masukkan lokasi Anda saat ini,
kemudian klik **Get Directions**
untuk menemukan arah ke tempat kami.

From address:

[Home](#) • [KONTAK](#) • [Peta Lokasi Kampus](#)

Dengan Peta ini Anda dapat mencari lokasi kampus Unisbank dari manapun Anda berada. Silahkan masukkan lokasi Anda saat ini pada **From Address** lalu klik **Get Directions**.