Perancangan Dan Pembuatan Pengamanan Video Chat Dengan Menggunakan Metode Enkripsi RC4

Stevie Suwanto Putra[1], Gregorius Satia Budhi[2], Justinus Andjarwirawan[3]
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-mail: steviesputra@gmail.com, greg@petra.ac.id[2], justin@petra.ac.id[3]

ABSTRAK: Perkembangan teknologi pada saat ini telah berkembang pesat dengan adanya internet. Seiring dengan perkembangan jaman makin banyak pula kegunaan internet. Tidak hanya digunakan untuk browsing, sekarang ini internet juga digunakan sebagai media komunikasi. Seperti email, chat, dan lain sebagainya. Makin berkembangnya teknologi komunikasi, maka orang mulai berpikir mengenai keamanan dari teknologi komunikasi tersebut. Seperti dalam video chat, hanya orang yang berhak atau yang berkepentingan saja yang dapat mengetahui informasi dalam video chat tersebut. Tetapi proses enkripsi pada proses yang sudah ada dikatakan seperti *blackbox* karena kurangnya transparasi.

Dari permasalahan tersebut memunculkan gagasan untuk membuat suatu aplikasi video chat yang memakai teknik pengamanan kriptografi. Data video chat akan dienkripsi memakai metode RC4. Penelitian menunjukan bila data yang telah terenkripsi dapat ditampilkan bila password benar. Hal ini menunjukan data yang telah terenkripsi lebih aman daripada data yang tidak dienkripsi..

Kata Kunci: *Video Chat*, RC4, Kriptografi.

ABSTRACT: *Technological development at this time has been growing rapidly with the internet . Along with the development the more the usefulness of the Internet . Not only used for browsing, now the Internet is also used for communication . Such as email, chat, and etc. The increasing development of communication technology, then people start to think about the secure of the communication technology. As in a video chat, only the person entitled to know the information in the video chat. But the encryption process on existing processes such as blackbox said due to lack of transparency.*

Of these problems lead to the idea to make a video chat application that uses cryptographic security techniques. Data of video chat will be encrypted using RC4 method. Research shows when the data that has been encrypted can be displayed when password is correct. It shows when data that has been encrypted safer than the data that is not encrypted.

***Keywords*:** *Video Chat, RC4, Cryptography.*

1. PENDAHULUAN

Perkembangan teknologi pada saat ini telah berkembang pesat dengan adanya internet. Seiring dengan perkembangan jaman makin banyak pula kegunaan internet. Tidak hanya digunakan untuk browsing, sekarang ini internet juga digunakan sebagai media komunikasi. Seperti *email*, *chat*, dan lain sebagainya.

Seiring dengan perkembangan teknologi komunikasi yang terus meningkat sampai sekarang ini makin berkembang pula teknologi *chat*. Dari *chat* yang hanya menggunakan kata – kata saja, sekarang berkembang dengan menggunakan video yang dikenal dengan teknologi video *chat*. Teknologi video *chat* memungkinkan seseorang berkomunikasi tatap muka tanpa harus bertemu. Jika merujuk pada aplikasi yang ada di pasaran sekarang, contoh aplikasi video *chat* adalah *Skype*, *Yahoo Messenger*. Makin berkembangnya teknologi komunikasi, maka orang mulai berpikir mengenai keamanan dari teknologi komunikasi tersebut. Seperti dalam video *chat*, hanya orang yang berhak atau yang berkepentingan saja yang dapat mengetahui informasi dalam video *chat* tersebut.

Salah satu teknik pengamanan pesan atau informasi yang dapat digunakan adalah kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan suatu berita atau informasi didalamnya dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Tujuan dari kriptografi adalah bagaimana pesan tersebut hanya dapat dibaca atau disampaikan kepada orang yang berhak untuk menerima pesan tersebut.

2. TINJAUAN PUSTAKA

2.1 Video

Video adalah gabungan dari banyak citra digital yang disertai dengan suara. Citra-citra digital tersebut diperlihatkan sesuai dengan urutan dengan jangka waktu tertentu sehingga gambar tampak bergerak. Untuk dapat mengolah video, maka harus mendapatkan data warna pada *frame – frame* yang ada pada video. Karena *frame – frame* tersebut berupa sebuah citra digital, maka dalam pengolahan video tidak dapat terlepas dari pengolahan digital dimana sesuai dengan algoritma sesuai format video yang dipakai, sebagai contoh adalah AVI, MPEG, H.264. [5]

2.2 Kriptografi

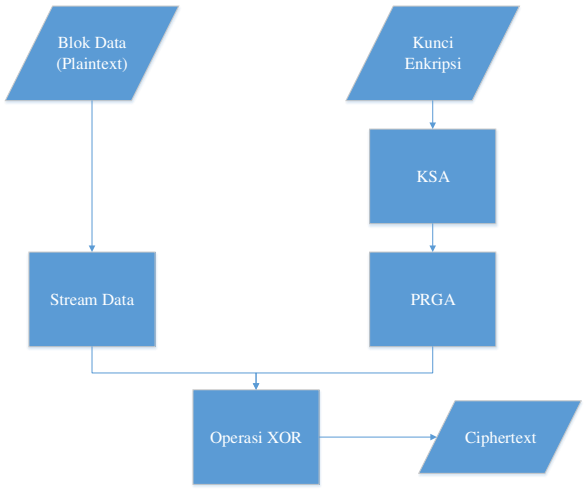
Kriptografi (*cryptography*) merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata *cryptós*

yang artinya *secret* (yang tersembunyi) dan *gráphein* yang artinya *writting* (tulisan).[4]

Jadi, kriptografi berarti *secret writting* (tulisan rahasia). Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology* yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah.Tetapi tidak semua aspek keamanan informasi ditangani oleh kriptografi.

2.3 RC4

RC4 merupakan salah satu jenis stream cipher yang didesain oleh Ron Rivest di labotarium RSA *(RSA Data Security Inc*) pada tahun 1987. RC4 sendiri merupakan kepanjangan dari Ron Code atau Rivest's Cipher. RC4 stream cipher ini merupakan teknik enkripsi dengan panjang kunci yang variabel dan beroperasi dengan orientasi byte. Algoritma RC4 terdiri atas 2 bagian yaitu *Key Scheduling Algorithm* (KSA) dan *Pseudo Random Generation* (PRGA) [3]. Dalam RC4 *key* yang digunakan untuk enrkripsi akan mengalami proses KSA dan PRGA yang kemudian akan diproses operasi XOR dengan *stream data* yang didapat. Hasil dari proses XOR tersebut adalah *ciphertext* atau data yang telah dienkripsi. Flowchart enkripsi RC4 dapat dilihat pada gambar 1.



Gambar 1: Flowchart Enkripsi RC4

2.4 Java

Java merupakan bahasa pemrograman tingkat tinggi yang dapat diterapkan pada banyak platform. Bahasa pemrograman java mempunyai ciri sebagai bahasa yang sederhana, arsitektur netral berorientasi obyek, mempunyai kinerja yang tinggi, *multithreaded*, kuat, dinamis dan aman. [2]

Java mempunyai kemampuan dapat berjalan di banyak *platform*. Sebuah *platform* adalah perangkat keras atau perangkat lunak lingkungan dimana program berjalan, seperti : Microsoft Windows, Linux, Solaris OS dan Mac OS. Platform java mempunyai dua komponen, yaitu *: Java Virtual Machine* dan *Java Application Programming Interface (API).*

2.5 *Java Media Framework*

JMF API merupakan arsitektur yang menggabungkan protokol dan pemrograman *interface* untuk merekam, mentransmisi, dan *playback* media. Pada JMF versi 2.1.1, Sun's sebagai perusahaan pengembang bahasa pemrograman java berinisiatif untuk membawa pemrosesan *time-base* media kedalam bahasa pemrograman Java. *Time-base* media adalah mengubah data yang diterima dengan berdasarkan waktu, termasuk didalamnya seperti audio dan video klip, MIDI, dan animasi. Konsep kerja JMF adalah seperti berikut. Sebuah *DataSource* meng-enkapsulasi media yang akan ditransmisikan seperti video tape dan player menyediakan mekanisme pemrosesan dan control sama seperti VCR (*Video Cassette Recorder*). Menjalankan dan merekam audio dan video dengan JMF membutuhkan peralatan input dan output yang tepat seperti mic, kamera, speaker, dan monitor.

DataSource dan *Player* adalah bagian integral dari API *(Application Programming Interface*) tingkat tinggi dari JMF untuk mengatur *capture*, presentasi, dan pemrosesan *time-based* media. JMF menyediakan *developer* Java dengan API yang mudah dipakai untuk mendukung *time-based* media ke dalam program Java, selama mempertahankan fleksibilitas dan ekstensibilitas yang dibutuhkan untuk mendukung aplikasi media tingkat tinggi.[6]

2.6 Pemrograman Jaringan Dengan Java

Pemrograman *socket* adalah cara untuk menggunakan komponen API (*Application Programming Interface*) socket untuk membuat sebuah aplikasi. Java telah menyediakan paket java.net yang berisi kelas-kelas dan *interface* yang menyediakan API (*Application Programming Interface*) level rendah (*Socket, ServerSocket, DatagramSocket*) dan level tinggi (*URL, URLConnection*). [1]

Socket merupakan fasilitas IPC *(Inter Proses Communication*) untuk aplikasi jaringan. Sebuah *socket* dilengkapi dengan alamat, yang terdiri atas IP *address* tujuan dan nomor *port*.
Alamat IP dapat menggunakan alamat jaringan lokal (LAN) maupun alamat internet. Jadi *socket* dapat digunakan untuk IPC pada LAN maupun Internet. Nomor *port* merupakan bilangan bulat yang digunakan untuk membedakan layanan-layanan yang berjalan pada komputer server yang sama. Pengguna layanan menggunakan nomor port ini untuk menghubungi komputer server dengan *workstation* (client). Dengan menggunakan nomor port yang standar, komunikasi dapat terjadi antar beberapa komputer dari jarak jauh untuk mengerjakan berbagai layanan jaringan, karena baik pengirim maupun penerima saling mengetahui ke mana data harus dikirim menggunakan nomor *port* tersebut. Sebagai contoh, semua sistem menggunakan nomor *port* 23 untuk aplikasi TELNET atau port 80 untuk aplikasi *website*. Oleh karena pada perancangan sistem akan dibuat suatu jenis layanan baru dengan memanfaatkan socket, maka dapat dibuat nomor *port* tersendiri untuk aplikasi tersebut.

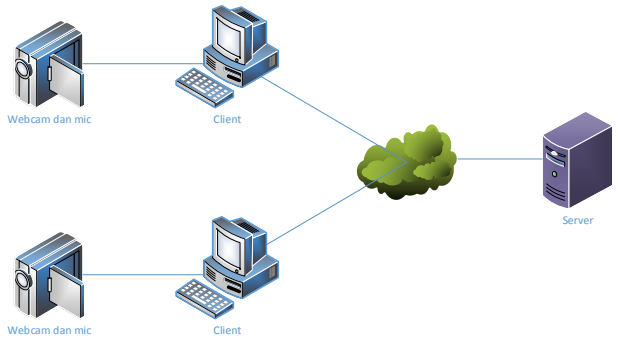
2.7 Freedom For Media In Java

FMJ adalah sebuah proyek *open source* dengan tujuan menyediakan alternatif untuk *Java Media Framework* (JMF), yang compatible dengan JMF. Hal ini bertujuan untuk menghasilkan *API framework* yang dapat digunakan untuk menangkap, pemutaran, proses, dan media *stream* di beberapa *platform*. [7]

3. DESAIN PROGRAM

3.1 *Network* Diagram

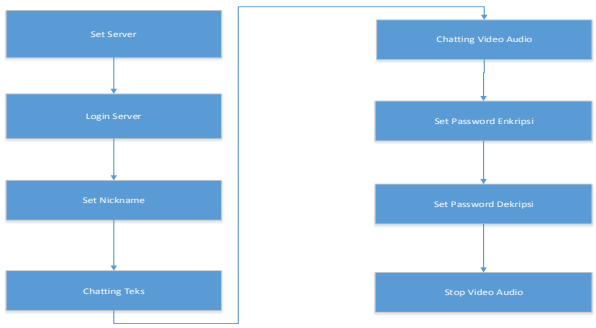
Network diagram menggambarkan bagaimana client dapat saling berinteraksi. Pengguna harus melakukan koneksi terlebih dahulu pada server sebelum melakukan proses video *chat*. Server akan menghubungkan antar pengguna. Saat proses video *chat* input yang dipakai untuk mengambil video dan audio adalah *webcam* dan *mic*. *Network diagram* dapat dilihat pada gambar 2.



Gambar 2: *Network* Diagram

3.2 Flowchart Program

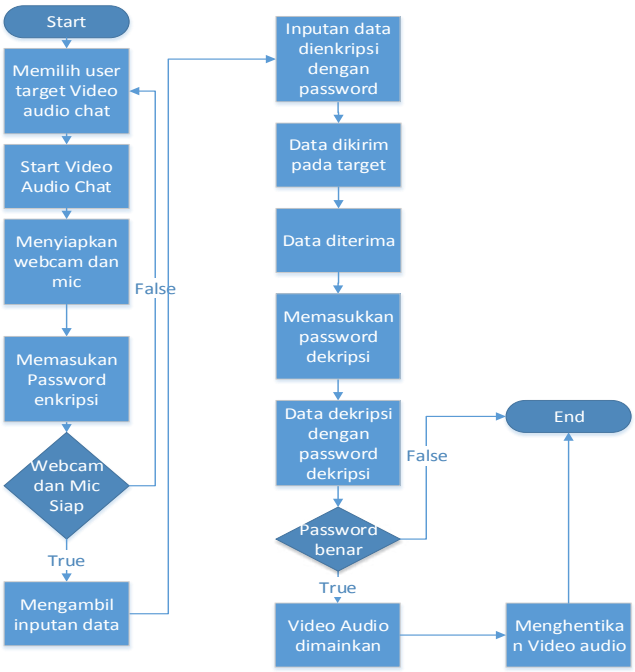
Sebelum memulai video *chat* pengguna harus melakukan login ke server terlebih dahulu. Setelah itu pengguna dapat menset *nickname* yang akan digunakan sehingga pengguna lainnya dapat mengenali. Kemudian pengguna dapat berinteraksi dengan pengguna lainnya dengan *chat* teks. Saat pengguna akan imelakukan video *chat*, pengguna dapat memakai video chat biasa maupun dapat menggunakan video *chat* yang telah dienkripsi yang tingkat keamanannya lebih baik daripada melakukan video *chat* secara biasa. Pengguna dapat memasukan *password* enkripsi dan dekripsi. Bila pengguna telah selesai chat video audio pengguna dapat *stop video audio chat*. Blok diagram inti program dapat dilihat pada gambar 3.



Gambar 3: Blok Diagram Inti Program

3.3 Proses Enkripsi Video Audio Chat

Inputan data dari *webcam* dan *mic* akan dienkripsi sebelum dikirim. Data tersebut akan dienkripsi menggunakan *password* enkripsi yang diminta oleh pengirim. Setelah data dienkripsi maka penerima akan diminta memasukan *password* dekripsi. Bila *password* dekripsi salah maka video atau audio tidak dapat ditampilkan. Sedangkan bila *password* dekripsi benar maka video atau audio akan ditampilkan. Flowchart proses enkripsi dapat dilihat pada gambar 4.



Gambar 4: Flowchart Proses Enkripsi

3.4 RC4

Proses enkripsi RC4 terbagi menjadi 2 bagian besar yaitu KSA (*Key Schedule Algorithm*) yang menghasilkan keadaan awal *array* S untuk kunci tertentu. Dan PRGA *(Pseudo Random Generator Algorithm)* yang menghasilkan *keystream byte* dari *array* s. Dalam KSA RC4 disiapkan array 'S' sebanyak 255 diisi dengan i. Kemudian proses permutasi S dengn cara 'J' ditambah dengan S[i]. Kemudian isi dari S[i] dengan S[j] ditukar. Setelah vektor s diinisialisasi, *key* enkripsi tidak diperlukan lagi. Dalam PRGA S[i] ditambah dengan j kemudian S[i] ditukar dengan S[j] dan hasil jumlah S[i] ditambah dengan S[j] *mod* 255 sebagai index K. Dimana K dipakai untuk proses XOR enkripsi. Gambar *pseudocode* KSA RC4 dapat dilihat pada gambar 5 dan gambar *pseudocode* PRGA RC4 dapat dilihat pada gambar 6.

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
```

```
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap values of S[i] and S[j]
endfor
```

Gambar 5: *Pseudocode* KSA RC4

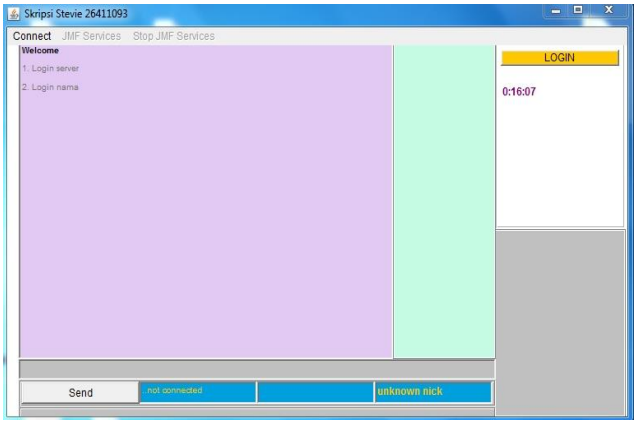
```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
endwhile
```

Gambar 6: Pseudocode PRGA RC4

4. Hasil Pengujian

4.1 Halaman Utama Program

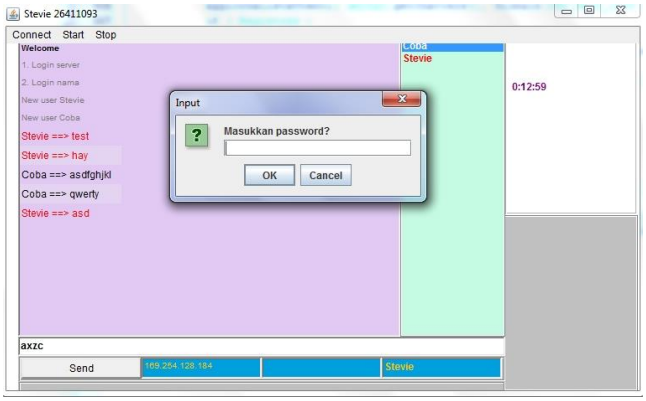
Hasil aplikasi adalah berupa program video chat. Gambaran aplikasi secara umum seperti pada Gambar 7.



Gambar 7: Tampilan Utama Program

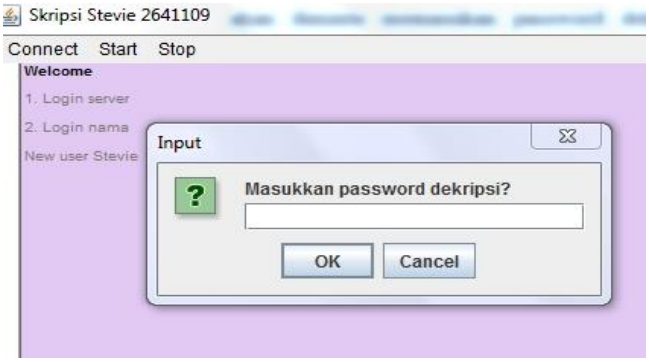
4.2 Enkripsi Video Audio *Chat*

Saat memulai enkripsi video audio *chat* pengirim akan diminta memasukan *password* enkripsi. Gambar permintaan password enkripsi dapat dilihat pada gambar 8.



Gambar 8: Password enkripsi

Kemudian pada penerima akan diminta memasukan *password* dekripsi. Password dekripsi akan digunakan untuk dekripsi data enkripsi yang diterima. Gambar permintaan password dekripsi dapat dilihat pada gambar 9.



Gambar 9: *Password* Dekripsi

Bila *password* yang dimasukan penerima sama dengan pengirim maka video audio *chat* dapat ditampilkan. Berikut gambar 10 bila password enkripsi dan password dekripsi benar.

Gambar 10: Audio Video *Chat* Pada PC A Dan PC B

Tetapi bila *password* dekripsi salah atau tidak sama dengan *password* enkripsi maka akan keluar *box warning* dan video audio *chat* tidak dapat ditampilkan. Gambar *box warning* dapat dilihat pada gambar 11.

Gambar 11: *Password* Dekripsi Salah

4.3 Video *Conference*

Video *Conference* dapat menampilkan sampai dengan 3 pengguna. Berikut adalah gambar saat 3 pengguna melakukan video conference dapat dilihat pada gambar 12.

Gambar 12: Hasil Video Conference PC A, PC B, PC C

4.4 Pengujian *Overhead*

Pengujian overhead dilakukan dengan pengujian CPU usage dan Memory. Pengujian dilakukan saat program belum dijalankan, program dijalankan, video audio chat tanpa memakai enkripsi, dan saat video audio chat memakai enkripsi. Berikut adalah hasil pengujian overhead dapat dilihat pada tabel 1.

Tabel 1 Pengujian *Overhead*

	CPU Usage	***Memory***
Program Belum dijalankan	17%	1.33 GB
Program Dijalankan	33%	1.71 GB
Video Audio Chat Tanpa Enkripsi	39%	1.46 GB
Video Audio Chat Enkripsi	43%	1.46 GB

Dari pengujian tabel 1 dapat disimpulkan bahwa bila ada proses yang lebih dijalankan maka *CPU usage* dan *memory* akan bertambah. Dan proses enkripsi tidak terlalu memakai *resource* yang banyak.

4.5 Pengujian Waktu Kirim

Pengujian waktu kirim dilakukan dengan menghitung waktu kirim dari PC A ke PC B. Pengujian dilakukan 5 kali percobaan. Hasil pengujian waktu kirim dapat dilihat pada tabel 2.

Tabel 2 Pengujian Waktu Kirim

Pengujian	**PC A Ke PC B (s)**
1	5.27
2	6.1
3	4.66
4	5.58
5	5.1

Dari tabel 2 dapat disimpulkan dibutuhkan waktu kurang lebih 5 detik untuk mengambil data input baik *webcam* maupun *mic* dan data dienkripsi.

5. Kesimpulan

- *User* dapat memiliki pilihan untuk berinteraksi dengan pengguna lainnya baik berupa audio *chat*, video *chat*, video audio *chat*.
- *Conference chat* dapat terjadi karena user yang dapat menggunakan program video *chat* ini sampai dengan 3 pengguna
- Keamanan lebih terjamin karena data yang dikirimkan berupa data enkripsi.
- Terdapat delay sekitar 5 detik untuk mempersiapkan *webcam* sampai data diterima pengguna penerima.
- Penambahan enkripsi menyebabkan *CPU usage* dan *memory* bertambah meskipun tidak terlalu tinggi.

6. Daftar Pustaka

[1] Deitel, Paul & Harvey Deitel. 2014. *Java SE8 for Programmers*. Prentice Hall.

[2] Ferguson, Niels, Bruce Schneier & Tadayoshi Kohno. 2011. *Cryptography Engineering Design Principles and Practical Applications*. Wiley.

[3] Kromodimoeljo, Sentot. 2010. *Teori dan Aplikasi Kriptografi*. Jakarta: SPK IT Consulting.

[4] Liguori, Robert & Patricia Liguori. 2014. *Java 8 Pocket Guide*.O'Reilly Media.

[5] Salter, David & Rhawi Dantas. 2014. *Netbeans IDE 8 Cookbook*. Packt Publishing.

[6] "*Penjelasan mengenai JMF*", http://www.cs.uccs.edu/~cs525/jmf/jmf.html, diakses pada tanggal 20 Agustus 2014.

[7] "*Penjelasan mengenai FMJ*", http://fmj-sf.net/, diakses pada tanggal 10 Oktober 2014