JURNAL DASI ISSN: 1411-3201
Vol. 11 No. 2 Juni 2010

PENERAPAN ALGORITMA BLOWFISH UNTUK MEMBUAT SEBUAH MODEL KRIPTOSISTEM ALGORITMA DAN MENGANALISIS KINERJA ALGORITMA BLOWFISH DENGAN SIMULASI DATA TERBATAS

Ema Utami, Shanty Erikawaty Aryani Tambunan
STMIK AMIKOM Yogyakarta

Abstraksi

Komunikasi via website memungkinkan terjadinya pengiriman berbagai varian paket data dalam jumlah yang besar. Data yang dikirim melalui media transmisi tersebut bisa saja disadap, dirusak atau dicuri oleh para cracker dan pihak-pihak yang memiliki kepentingan tertentu. Masalah cyber crime ini masih menjadi masalah yang sulit dikendalikan hingga saat ini. Untuk mengatasi masalah keamanan data ini, penulis melakukan pendekatan teknologi enkripsi data menggunakan algoritma Blowfish, enkripsi yang termasuk dalam golongan Symmetric Cryptosystem.

Kata kunci: *algoritma Blowfish, enkripsi, kriptosistem*

Pendahuluan

Cryptosystem

Menurut teknik enkripsinya, *Cryptosystem* dapat digolongkan menjadi dua macam (Idocisc, 2005), yaitu :

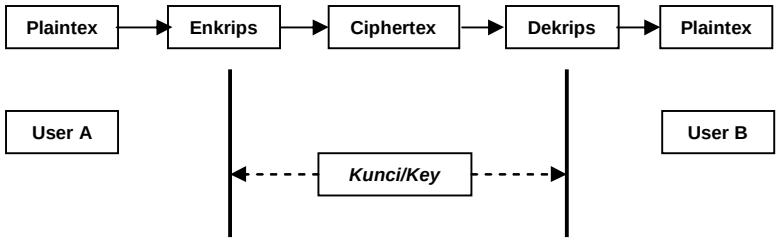
Symmetric Cryptosystem (Enkripsi Konvensional)

Kunci yang digunakan dalam proses enkripsi dan dekripsi adalah sama atau pada prinsipnya identik (Stallings, 1995). Kunci ini pun bisa diturunkan dari kunci lainnya. Sistem ini sering disebut juga dengan *secret-key ciphersystem*.

Jumlah kunci yang dibutuhkan umumnya adalah :

$$ {}_nC_2 = \frac{n*(n-1)}{2} $$

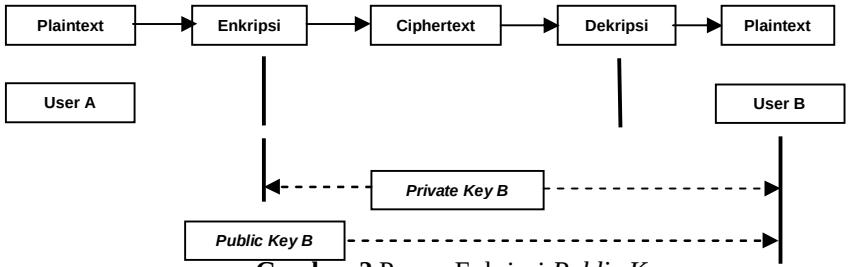
Dimana *n* adalah banyaknya pengguna. Kunci yang menggunakan teknik enkripsi ini harus betul-betul dirahasiakan. Gambaran proses enkripsi konvensional :



Gambar 1 Proses Enkripsi Konvensional

***Assymmetric Cryptosystem* (enkripsi *public-key*)**

Kunci yang digunakan terdapat dua buah, yaitu: kunci yang dapat dipublikasikan (*public key*) dan kunci yang harus dirahasiakan atau disebut kunci privat (*private key*). Secara sedehana proses tersebut diterangkan sebagai berikut: A mengirimkan pesan kepada B. A menyandikan pesannya dengan menggunakan kunci publik B. Bila B ingin membaca pesan dari A maka harus menggunakan kunci privatnya untuk mendekripsikan pesan yang tersandikan itu. Gambaran proses enkripsi *public-key* :



Gambar 2 Proses Enkripsi *Public Key*

Dalam kriptosistem yang baik harus memiliki karakteristik sebagai berikut (Syafari, 2003):

a. Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang dipergunakan.
b. Memiliki ruang kunci (*keyspace*) yang besar.
c. Menghasilkan *ciphertext* yang terlihat acak dalam seluruh tes statistik yang dilakukan terhadapnya.
d. Mampu menahan seluruh serangan yang dikenal sebelumnya.

Jika satu atau beberapa dari karakteristik di atas tidak dimiliki oleh sebuah kriptosistem, maka kemungkinan besar kriptosistem tersebut akan mudah mengalami gangguan dari pihak luar.

Algoritma Blowfish

Blowfish atau disebut juga *OpenPGP.Cipher.4* adalah enkripsi yang termasuk dalam golongan *Symmetric Cryptosystem* (Schneier, 1993). Blowfish dibuat untuk digunakan pada komputer yang mempunyai mikroprosesor besar (32 bit ke atas dengan cache data yang besar). Blowfish merupakan cipher blok yang berarti selama proses enkripsi dan dekripsi, Blowfish bekerja dengan membagi pesan menjadi blok-blok bit dengan ukuran sama panjang, yaitu 64-bit dengan panjang kunci bervariasi yang mengenkripsi data dalam 8 byte blok. Pesan yang bukan merupakan kelipatan 8 byte akan ditambahkan bit-bit tambahan (*padding*) sehingga ukuran untuk tiap blok sama. Algoritma Blowfish terdiri dari dua bagian: key expansion dan enkripsi data (Schneier, 1993).

1. Key expansion berfungsi untuk mengkonversikan sebuah kunci sampai 56 byte (448 bit) menjadi beberapa array subkey dengan total 4168 byte.
2. Enkripsi data, proses ini terjadi di dalam jaringan feistel dan terdiri dari iterasi fungsi sederhana sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi key-dependent serta substitusi kunci dan data-dependent. Semua operasi merupakan XOR dan

penjumlahan (*addition*) pada variable 32 bit. Operasi penambahan yang terjadi hanya merupakan empat indeks array data lookup pada setiap iterasi

Enkripsi Algoritma Blowfish

Blowfish menggunakan subkunci besar yang harus dihitung sebelum enkripsi dan dekripsi data. Algoritma Blowfish menerapkan jaringan Feistel yang terdiri dari 16 putaran. Input adalah elemen 64-bit, X untuk alur algoritma enkripsi dengan metode Blowfish dijelaskan sebagai berikut (Schneier, 1993):

1. Bentuk inisial P-array sebanyak 18 buah (P1, P2, ..., P18) masing-masing bernilai 32-bit. Array P terdiri dari delapan belas kunci 32-bit subkunci: P_1,P_2, ..., P_{18}
2. Bentuk S-box sebanyak 4 buah masing-masing bernilai 32-bit yang memiliki masukan 256. Empat 32-bit S-box masing-masing mempunyai 256 entri :
   $S_{1,0}$, $S_{1,1}$, ..., $S_{1,255}$
   $S_{2,0}$, $S_{2,1}$, ..., $S_{2,255}$
   $S_{3,0}$, $S_{3,1}$, ..., $S_{3,255}$
   $S_{4,0}$, $S_{4,1}$, ..., $S_{4,255}$
3. Plaintext yang akan dienkripsi diasumsikan sebagai masukan, Plaintext tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR.
5. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR.
6. Hasil dari operasi diatas ditukar XL menjadi XR dan XR menjadi XL.
7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
8. Pada proses ke-17 lakukan operasi untuk XR = XR xor P17 dan XL = XL xor P18.

9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-bit kembali.

Fungsi F adalah sebagai berikut:

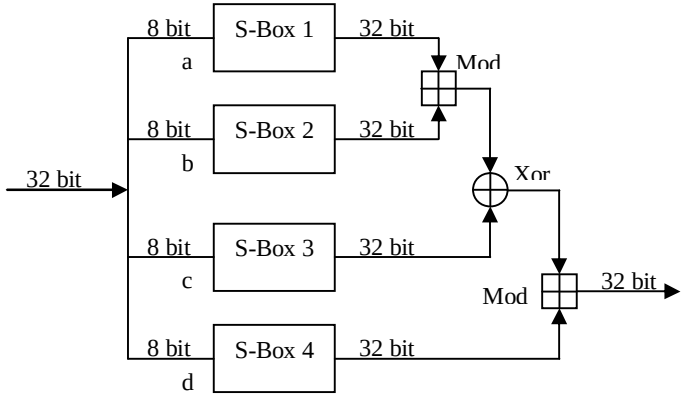
Bagi XL menjadi empat bagian 8-bit: a, b, c, d.

$$F(x_L) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) \oplus S_{3,c}) + S_{4,d} \bmod 2^{32}$$

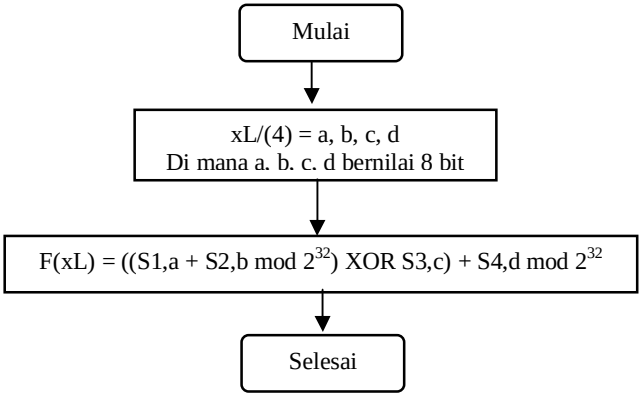
Subkunci dihitung menggunakan algoritma Blowfish, metodenya adalah sebagai berikut (Schneier, 1993):

1. Pertama-tama inilialisasi P-array dan kemudian empat S-box secara berurutan dengan string yang tetap. String ini terdiri atas digit hexadesimal dari Pi.
2. XOR P1 dengan 32-bit pertama kunci, XOR P2 dengan 32-bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai P18).Ulangi terhadap bit kunci sampai seluruh P-array di XOR dengan bit kunci.
3. Enkrip semua string nol dengan algoritma Blowfish menggunakan subkunci seperti dijelaskan pada langkah (1) dan (2).
4. Ganti P1 dan P2 dengan keluaran dari langkah (3).
5. Enkrip keluaran dari langkah (3) dengan algoritma Blowfish dengan subkunci yang sudah dimodifikasi.
6. Ganti P3 dan P4 dengan keluaran dari langkah (5).
7. Lanjutkan proses tersebut, ganti seluruh elemen dari P-array, kemudian seluruh keempat S-box berurutan, dengan keluaran yang berubah secara kontinyu dari algoritma Blowfish.

Berikut dapat dilihat gambar blok diagram dari fungsi F:



Gambar 3 Fungsi F (Bruce Schneier, 1996)



Gambar 4 Flowchart Fungsi F

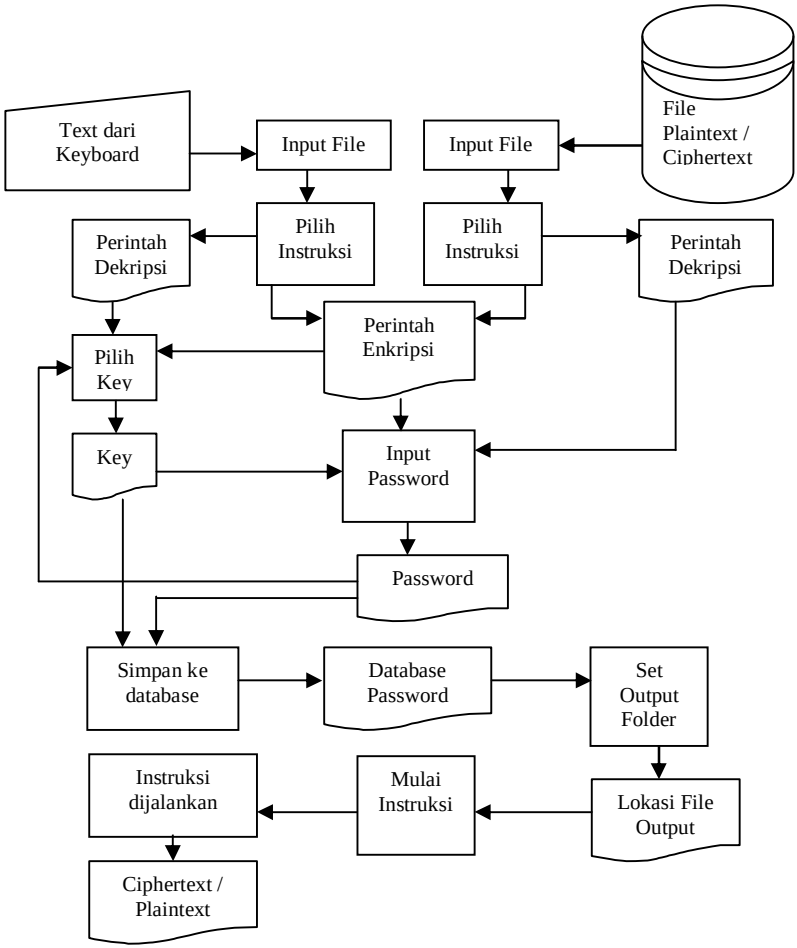
Dalam fungsi F terdapat total 521 iterasi untuk menghasilkan semua subkunci yang dibutuhkan. Aplikasi kemudian dapat menyimpan subkunci ini dan tidak membutuhkan langkah-langkah proses penurunan berulang kali, kecuali kunci yang digunakan berubah.

Pembahasan

Perancangan Sistem

Data yang diinputkan pada sistem bisa berasal dari dua sumber, yaitu dari harddisk dan dari masukan melalui keyboard ke textbox. Setelah proses input file, maka user harus memilih instruksi yang akan digunakan untuk memproses file yang telah diinputkan (enkripsi/dekripsi). Untuk pilihan proses enkripsi, maka pengguna diminta memilih *key* yang akan digunakan untuk enkripsi/dekripsi data (kunci:simetris), kemudian memasukkan password untuk *privacy user* yang mana password dan kunci tiap *user* akan disimpan ke dalam database sistem agar dapat digunakan lagi. Setelah itu, *user* harus menentukan lokasi untuk file output dan instruksi dapat dijalankan sehingga menghasilkan file ciphertext yang akan langsung dicopy ke directory/folder tujuan atau file output hanya akan ditampilkan pada *textbox* jika file input merupakan text masukan pada *textbox* juga.

Untuk dekripsi, proses yang terjadi adalah sama dengan proses enkripsi. Hanya saja jika file yang akan didekripsi merupakan file yang sudah dienkripsi sistem, maka user tidak diminta memilih kunci lagi. Karena kunci dekripsi akan menggunakan kunci yang sama saat proses enkripsi dan file output dari proses dekripsi adalah merupakan file plaintext.



Gambar 5 Flowchart Sistem

Pengujian

Pengujian sistem dilakukan dengan menerapkan tujuh prinsip yang umum digunakan untuk mendasari pengujian sebuah perangkat lunak: operabilitas, observabilitas, kontrolabilitas, dekomposabilitas,

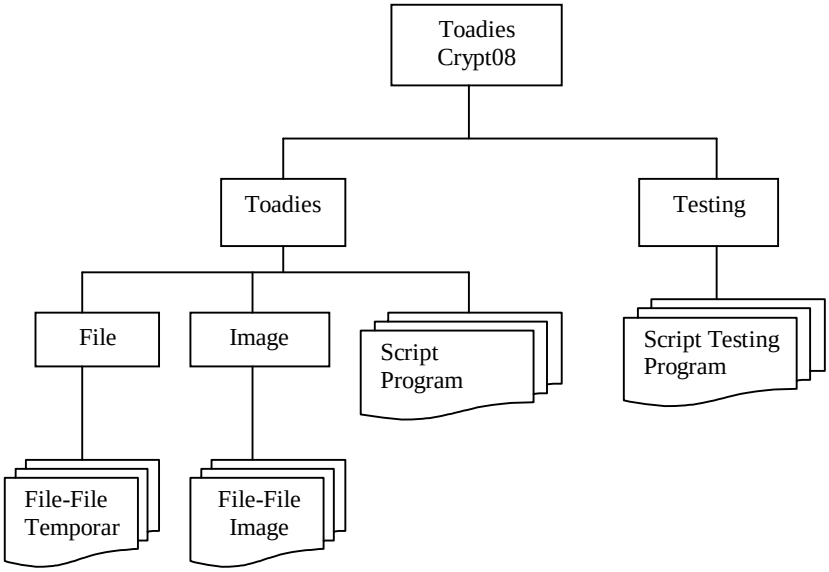
kesederhanaan, stabilitas, dan mudah dipahami. Bug pada *Toadies Crypt08* antara lain sebagai berikut:

1. Fungsi e*ncrypt* berfungsi untuk melakukan eksekusi enkripsi terhadap data bertipe string. Bug pada fungsi ini dapat dilihat pada baris ke empat fungsi, yaitu *Plain text must be a string*, berarti jika masukan plaintext terhadap fungsi ini tidak bertipe string, program akan menjalankan bug tersebut.
2. Fungsi *decrypt* berfungsi untuk melakukan eksekusi dekripsi terhadap data terenkripsi. Jika masukan *ciphertext* tidak sesuai dengan permintaan, maka program akan menjalankan bugnya, yaitu *Chipertext must be a string*.
3. Fungsi *setKey* memiliki parameter berupa key. Bug fungsi ini adalah *Key must be a string* dan *Key must be less than 56 characters and non-zero.* Ini berarti bug untuk tipe data key harus string dan batasan untuk panjang karakter key adalah bukan nol dan maksimal 56 karakter.
4. Fungsi *maxi_pad* berfungsi untuk memberikan masukan karakter pada karakter kosong dalam sebuah blok dengan pola bit tertentu (proses *padding*). Jika karakter masukan, dalam hal ini adalah *$plaintext*, habis dibagi delapan maka proses *padding* tidak dijalankan.

Semakin baik mengontrol perangkat lunak dapat dilakukan, semakin banyak pengujian yang dapat diotomatisasi dan dioptimalkan. Hal ini dapat dilihat dari apakah format input dan output konsisten dan terstruktur. Untuk mode operasi CBC dalam program, enkripsi string menggunakan enkripsi dasar *base64_encode* dan base64_decode. Sehingga hasil *ciphertext*-nya konsisten dan terstruktur.

Dengan mengontrol ruang lingkup pengujian, masalah dapat dengan lebih cepat diisolas dan pengujian sistem dapat dilakukan kembali secara lebih halus. Sistem dibangun dari modul yang independen dan dapat diuji secara independen. Hasil pengujian sistem dalam modul text maupun file, enkripsi maupun dekripsi, kecepatan kerja sistem sangat dipengaruhi oleh kecepatan kerja prosesor dan besar memory komputer. Kesederhanaan di sini berarti, semakin

sedikit yang diuji, semakin cepat sistem dapat diuji. Kesederhanaan fungsi, struktur dan kode. Variabel pengujian masih relatif sederhana, karena hanya terdiri dari: key, plaintext, dan ciphertext. Struktur dan kode program relatif sederhana. Gambaran struktur file sistem adalah sebagai berikut:



Gambar 6 Struktur File Sistem

Stabilitas dapat dilihat dari semakin sedikit perubahan, maka semakin sedikit gangguan dalam pengujian. Tujuan sistem sebagai model kriptosistem telah dicapai dengan berjalannya proses enkripsi dan dekripsi text dan file. Pengembangan yang dapat dilakukan dari sisi hardware adalah dengan melakukan implementasi sistem pada spesifikasi hardware yang lebih tinggi. Hal ini adalah untuk peningkatan kapasitas *plaintext* yang dapat dienkripsi. Semakin besar kapasitas file yang dapat dienkripsi, maka sistem akan semakin *useful* untuk diterapkan dalam berbagai macam data, sedangkan dari sisi

software, peningkatan pada logika program dan sistem dapat dimanipulasi lagi sesuai dengan keinginan pengembang, antara lain:

1. File algoritma utama, yaitu Blowfish.php dapat disederhanakan menjadi satu dengan file mode operasi yang digunakan.
2. Enkripsi dasar yang diterapkan dapat diganti dengan selain *base64_encode*.
3. Bisa dikembangkan dengan adanya lebih banyak *pop-up* yang lebih informatif.

Penutup

Implementasi algoritma Blowfish yang optimal dapat dilakukan dengan aplikasi yang tidak sering berubah-ubah kunci serta tidak menggunakan *weak-key*. Dalam fungsi F terdapat total 521 iterasi untuk menghasilkan semua subkunci yang dibutuhkan. Aplikasi kemudian dapat menyimpan subkunci ini dan tidak membutuhkan langkah-langkah proses penurunan berulang kali, kecuali kunci yang digunakan berubah. Kunci yang sering berubah akan membutuhkan proses penurunan baru pada iterasi yang panjang, hal ini akan membuat waktu kerja Blowfish lebih panjang, sedangkan penggunaan *weak key* dapat mengganggu hasil enkripsi dan dekripsi. *Weak key* membuat hasil enkripsi/dekripsi menjadi tidak konsisten. Tingkat keamanan algoritma Blowfish ditentukan oleh jumlah iterasi dan panjang serta kerahasiaan kunci yang digunakan jumlah iterasi yang digunakan semestinya membuat jaringan feistel pada Blowfish bekerja semestinya (16 iterasi), pengurangan jumlah iterasi akan mengurangi tingkat kesulitan suatu data untuk dipecahkan, sedangkan peran panjang dan kerahasiaan kunci menjadi sangat krusial. Kunci yang panjang menjadi sama tingkat kebutuhannya dengan iterasi yang tidak dikurangi karena proses pembangkitan *sub key* akan menjadi lebih acak dan membutuhkan waktu lama untuk dipecahkan. Begitu juga dengan kerahasiaan kunci. Jika kunci sudah diketahui oleh pihak yang tidak berkepentingan, maka akan sangat mudah memecahkan *ciphertext* atau *plaintext* tanpa *attack* tertentu sekalipun..

Daftar Pustaka

Schneier, Bruce. Applied Cryptography : Protocols, Algorithms, and Source Code in C. USA, John Wiley & Sons, Inc., 1996

Schneier. Bruce. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) (http://schneier.com. Diakses 13 januari 2008)

Rhee, Man Young. *Cryptography and Sesure Communications*, Singapore, McGraw-Hill Book Co.1994

Menezes, Alfred J. Paul C. van Oorschot and Scott A. Vanstone. *Handbook of Applied Cryptography* (http://id.wikipedia.org/wiki/Kriptografi. Diakses 21 Maret 2008)

Sunarfrihantono, Bimo. *PHP dan MySQL untuk Web*, Andi Offset, Yogyakarta. 2002

Ratih, *Studi dan Implementasi Algoritma Blowfish Untuk Aplikasi Enkripsi dan Dekripsi File* (www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-077.pdf. Diakses 14 Februari 2008)

Syafari, Anjar. *Sekilas tentang enkripsi Blowfish (*http://ilmukomputer.com/wp-content/uploads/2007/07/anjar-enkripsi-blowfish.doc. Diakses 3 Maret 2008)

Supani, Ahyar. *Sistem keamanan File dan Folder Data Menggunakan algoritma Blowfish dengan Kunci Simetrik* (www.cert.or.id/~budi/ courses/ el695/projects/report-ahyar.doc. Diakses 3 Maret 2008)

Jogiyanto. *Analisis & Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Andi Offset, Yogyakarta. 2005

Indocisc *Pengantar Kriptografi* (http://www.cert.or.id/~budi/courses/ec5010/04-kriptografi.pdf. Diakses 9 Februari 2008)

Schneier B. *Mode Of DES* (www.di-mgt.com.au Diakses 4 Maret 2008)

Sommerville, Ian. *Software Engineering 7th Edition.* Lancaster University. 2004