ANALISA IMPLEMENTASI ALGORITMA STREAM CIPHER SOSEMANUK DAN DICING DALAM PROSES ENKRIPSI DATA

Endro Ariyanto[1], Trisya Indah Pravitasari[2], Setyorini[3]
[1,2,3]Departemen Teknik Informatika Institut Teknologi Telkom, Bandung
Jl. Telekomunikasi No.1 Dayeuhkolot, Bandung (022-7564108)
[1]end@stttelkom.ac.id, [2]cah_sya@yahoo.com, [3]srn@stttelkom.ac.id

Abstrak
Keamanan merupakan hal yang diutamakan dalam sistem informasi, khususnya dalam pertukaran data yang bersifat penting atau rahasia. Informasi yang akan diberikan kepada pihak yang berhak terhadap informasi tersebut harus benar-benar dijaga tingkat keamanannya, jangan sampai jatuh ke tangan pihak lain yang tidak punya hak akan informasi tersebut.
Salah satu cara untuk menjaga keamanan informasi yang dipertukarkan dalan suatu sistem dapat dilakukan dengan menggunakan teknik kriptografi. Kriptografi merupakan seni dan ilmu untuk menyembunyikan informasi dari pihak ketiga. Dalam kriptografi seseorang yang memiliki kunci privat dapat mengubah data asli (plaintext) menjadi data yang bersifat unik dan tidak dapat dibaca (ciphertext) dan dapat mengubah kembali ciphertext yang ada ke dalam bentuk plaintext dengan menggunakan kunci privat yang dimilikinya.
Dalam penelitian ini telah berhasil dibuat suatu sistem kriptografi menggunakan algoritma Sosemanuk dan Dicing dan diimplementasikan menggunakan Borland C++ Builder 6.0. Penelitian ini bertujuan untuk menganalisa perbandingan performansi antara algoritma Sosemanuk dan Dicing dalam hal kecepatan proses enkripsi dan dekripsi, memori yang dibutuhkan selama proses, dan nilai avalanche effect.
Berdasarkan hasil penelitian dapat disimpulkan bahwa nilai Avalanche Effect (AV) algoritma Sosemanuk lebih besar daripada algoritma Dicing, sehingga algoritma Sosemanuk lebih handal daripada algoritma Dicing. Proses pada algoritma Sosemanuk lebih kompleks daripada algoritma Dicing, sehingga waktu yang diperlukan oleh algoritma Sosemanuk 4,77 % lebih lama dan memori yang diperlukan lebih besar daripada algoritma Dicing. Tipe file tidak berpengaruh terhadap lama waktu enkripsi ataupun dekripsi, karena file dibaca per byte.

***Keyword**: kriptografi, kunci privat, Sosemanuk, Dicing*

1. PENDAHULUAN

Dewasa ini penggunaan komputer untuk pengiriman data melalui saluran komunikasi sudah merupakan hal yang jamak. Namun sekarang ini banyak orang yang tidak bertanggung jawab dengan melakukan sabotase terhadap pengiriman data melalui jaringan. Hal ini mengakibatkan perlu adanya tingkat keamanan yang lebih baik dalam hal pengiriman data lewat jaringan. Untuk mencegah adanya penyadapan data pada waktu pengiriman, digunakanlah teknik kriptografi untuk menyandikan data, yaitu dengan cara mengubah teks asli (*plaintext*) menjadi teks yang tersandi (*ciphertext*) yang tidak mempunyai makna dan tidak dapat dibaca.
Dengan teknik kriptografi yang menggunakan proses enkripsi dan dekripsi, maka suatu data dapat diubah ke bentuk yang tidak dimengerti oleh orang awam dan dapat dikembalikan lagi ke bentuk data semula. Enkripsi adalah proses mengubah suatu data asli (*plaintext*) ke dalam bentuk yang tidak dapat dibaca (*ciphertext*) dengan menggunakan suatu kunci. Sedangkan dekripsi adalah proses untuk mengubah data yang sudah berupa data yang tidak dapat dibaca (*ciphertext*) kembali ke bentuk data asli (*plaintext*) dengan menggunakan suatu kunci. Terdapat dua jenis kunci yaitu kunci simetri dan kunci asimetri. Pada kunci simetri digunakan sebuah kunci yang sama dalam proses enkripsi dan dekripsi, sedangkan pada kunci asimetri digunakan dua kunci berbeda yaitu kunci pribadi untuk melakukan dekripsi dan kunci publik untuk melakukan enkripsi. Kunci simetri memiliki dua macam algoritma. Algoritma pertama adalah *stream cipher* yaitu algoritma yang beroperasi pada *plaintext* yang berupa satu bit/byte tunggal pada satu kurun waktu. Algoritma yang kedua adalah *block cipher* yaitu algoritma yang beroperasi pada *plaintext* dalam kelompok bit-bit yang disebut blok.
Pada penelitian ini dilakukan analisis terhadap hasil implementasi antara algoritma Sosemanuk dan algoritma Dicing dengan mencari perbedaan di antara keduanya. Algoritma Sosemanuk dan Dicing merupakan algoritma *stream cipher* terbaru yang memiliki panjang kunci antara 128 sampai 256 bit. Saat ini algoritma Sosemanuk hanya menjamin keamanan kuncinya pada 128 bit. Algoritma Sosemanuk memiliki *initial vector* 128 bit sedangkan *initial vector* algoritma Dicing tergantung pada panjang kunci yang digunakan. Kedua algoritma *stream cipher* ini menggunakan kombinasi LFSR (*Linear Feedback Shift Register*) dan FSM (*Finite State Machine*) sebagai pembangkit kuncinya.
Permasalahan yang dijadikan objek penelitian dan pengembangan adalah:

1. Bagaimana membangun aplikasi dengan menerapkan algoritma Sosemanuk dan algoritma Dicing.
2. Bagaimana memperoleh nilai kecepatan proses, jumlah memori yang digunakan, dan nilai avalanche effect pada saat enkripsi dan dekripsi data pada algoritma Sosemanuk dan algoritma Dicing.

3. Bagaimana kualitas file yang telah mengalami proses enkripsi dan dekripsi pada algoritma Sosemanuk dan algoritma Dicing.

Berdasarkan permasalahan di atas, maka pada penelitian ini dilakukan implementasi algoritma Sosemanuk dan Dicing ke dalam sebuah aplikasi enkripsi dan dekripsi data serta analisis terhadap hasil-hasil yang telah diperoleh.

2. TINJAUAN PUSTAKA

2.1 *Linear Feedback Shift Register*

Linear Feedback Shift Register (LFSR) sering digunakan oleh *Stream cipher* sebagai pembangkit *keystream*. Register geser umpan-balik atau *Feedback Shift Register (FSR)* terdiri dari dua bagian: (Kurniawan, 2004)

1. Register Geser, yaitu barisan bit-bit yang panjangnya n-bit. Register geser disebut juga sebagai register geser n bit.
2. Fungsi Umpan-balik, yaitu fungsi yang menerima masukan dari register geser dan mengembalikan nilai fungsi ke register geser.

2.2 Avalanche effect

Avalanche Effect adalah perubahan satu bit pada *plaintext* atau *key* yang menyebabkan perubahan yang signifikan terhadap *ciphertext.* Suatu algoritma dikatakan memiliki nilai AE yang baik jika perubahan satu bit saja pada *input* menghasilkan perubahan sekitar setengah jumlah bit pada *output*-nya. Maksimal nilai AE adalah 50% (Wikipedia, 2008). Salah satu fungsi dari AE adalah untuk melihat tingkat keamanan suatu algoritma kriptografi.

$$Avalanche\_Effect(AE) = \frac{\sum bit\_berubah}{\sum bit\_total} * 100\% \quad (1)$$

2.3 Stream cipher

Stream cipher digunakan untuk mengenkripsi *plaintext* menjadi *ciphertext* bit per bit (1 bit setiap kali transformasi) atau *byte* per *byte* (1 karakter = 1 *byte*). *Stream cipher* pertama diperkenalkan oleh Vernam yang diadopsi dari *one-time pad cipher*, yaitu tiap karakter diganti dengan bit 0 atau 1 (Kurniawan, 2004).

Ciphertext diperoleh dengan rumus:

$$c_i = p_i \oplus k_i \quad (2)$$

Sedangkan proses dekripsi diperoleh dengan rumus:

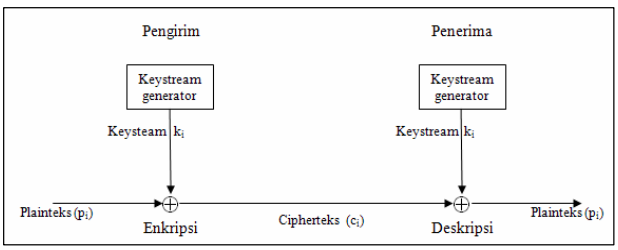
$$p_i = c_i \oplus k_i \quad (3)$$

dalam hal ini, p_i : bit *plaintext*
c_i : bit *ciphertext*
k_i : bit kunci

Pada *stream cipher,* bitnya hanya memiliki dua nilai yaitu berubah atau tidak berubah yang ditentukan oleh kunci enkripsi yang disebut *keystream. Keystream* dibangkitkan oleh sebuah pembangkit yang dinamakan *keystream generator* (pembangkit kunci). *Keystream* di-XOR-kan dengan *plaintext* menghasilkan *ciphertext* (rumus 2). Di sisi penerima, bit-bit *ciphertext* di-XOR-kan dengan bit-bit kunci yang sama untuk menghasilkan *plaintext* kembali, karena:

$$c_i \oplus k_i = (p_i \oplus k_i) \oplus k_i = p_i \oplus (k_i \oplus k_i) = p_i \oplus 0 = p_i \quad (4)$$

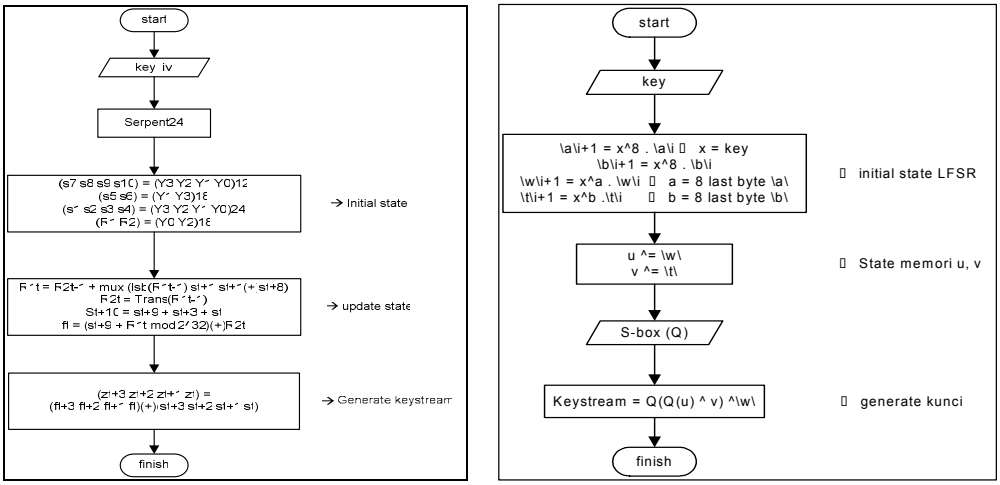
Gambar 1 memperlihatkan konsep *stream cipher*, di mana pembangkit kunci menghasilkan bit kunci k_i yang kemudian di-XOR-kan dengan *plaintext* p_i menghasilkan bit *ciphertext* c_i. Di sisi penerima pembangkit kunci yang sama akan meng-XOR-kan bit *ciphertext* c_i dengan kunci k_i yang sama untuk menghasilkan *plaintext* p_i awal.



Gambar 1: Konsep stream cipher

2.4 Pembangkit kunci

Proses yang terjadi dalam pendefinisian kunci menggunakan algoritma Sosemanuk dijabarkan pada gambar 2a, sedangkan proses pada algoritma Dicing dijabarkan pada gambar 2b.

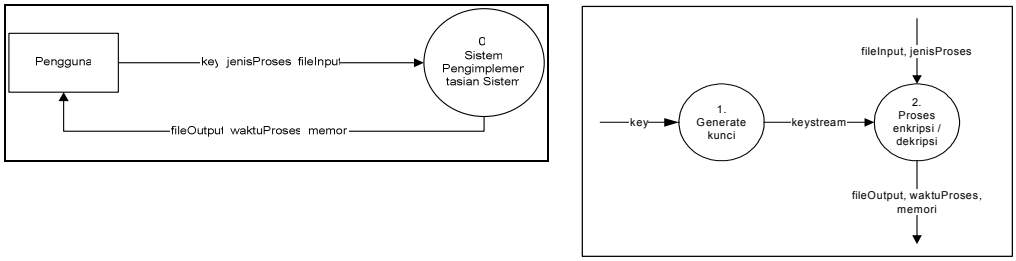


(a) (b)

Gambar 2: Proses pendefinisian kunci (a) algoritma Sosemanuk (b) algoritma Dicing

3. METODE PENELITIAN

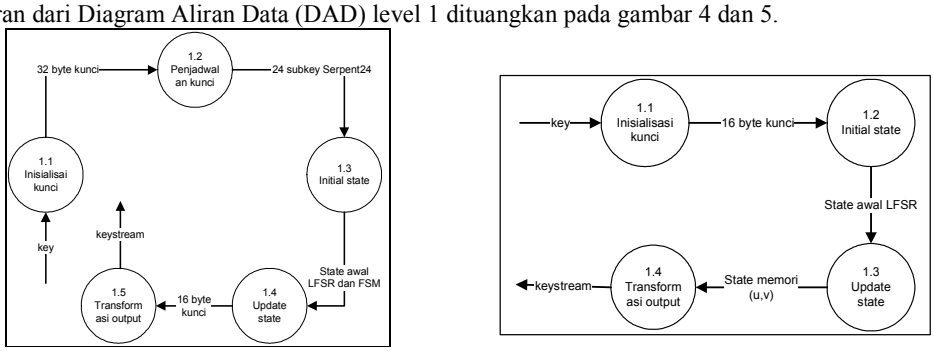
Gambaran umum dari sistem yang dibangun dapat dilihat pada gambar 3a. Sistem akan meminta pengguna untuk memasukkan kunci privat, jenis proses, file input dan file output untuk melakukan proses enkripsi atau dekripsi data. Sistem akan membentuk *keystream* dari kunci privat yang dimasukkan dan melakukan operasi bitwise XOR dengan *plaintext* sehingga menghasilkan *ciphertext*. Selain menghasilkan *ciphertext*, sistem juga akan mengeluarkan informasi mengenai lama waktu proses (ms), besar file (byte) dan memori yang digunakan (kB).



(a) (b)

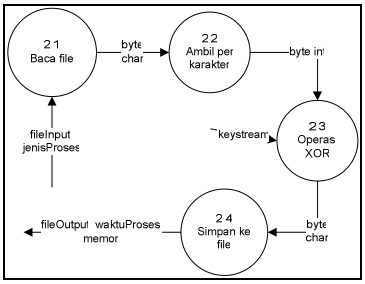
Gambar 3: (a) DAD level 0 (b) DAD level 1

Penjabaran dari Diagram Aliran Data (DAD) level 1 dituangkan pada gambar 4 dan 5.



(a) (b)

Gambar 4: DAD level 2 (a) Generate kunci Sosemanuk (b) Generate kunci Dicing



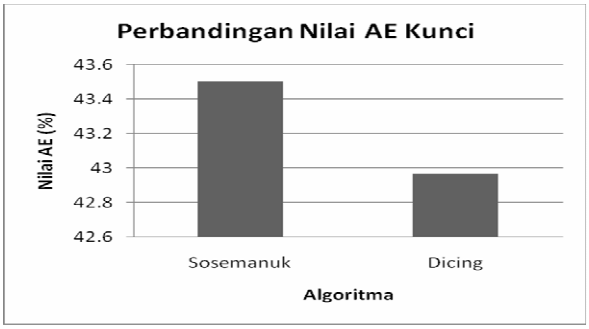
Gambar 5: DAD level 2 Enkripsi/Dekripsi

Parameter yang digunakan dalam perbandingan antara algoritma Sosemanuk dan algoritma Dicing adalah:

1. Avalanche Effect (AE)
   Untuk menentukan nilai AE dapat dilakukan dengan dua cara yaitu menggunakan dua buah kunci yang berbeda satu bit pada sebuah *plaintext* untuk menghasilkan *ciphertext*. Pengujian ini dilakukan pada file teks dengan membandingkan jumlah bit yang berbeda pada *ciphertext* yang dihasilkan dan seberapa besar perbedaan yang diakibatkan satu bit tersebut pada AE. Kunci yang digunakan adalah "aekey2" dan "aekey3", sedangkan plaintext yang digunakan adalah "aku anak nomer 1".
2. Lama waktu proses
   Lama waktu proses diukur untuk membandingkan kecepatan enkripsi dan dekripsi antara algoritma Sosemanuk dan algoritma Dicing untuk melihat berapa besar perbedaan kecepatan pemrosesan data antara kedua algoritma. Pengujian dilakukan terhadap empat file dengan ukuran dan jenis yang berbeda yaitu dua buah file text (coba1.txt berukuran 646.665 byte dan coba2.txt berukuran 1.077.144 byte) dan dua buah file gambar (coba3.bmp berukuran 3.240.056 byte dan coba4.bmp berukuran18.289.208 byte).
3. Memori yang dipakai
   Memori yang digunakan diukur untuk melihat berapa banyak memori yang diperlukan selama proses enkripsi dan dekripsi data antara algoritma Sosemanuk dengan algoritma Dicing.

4. HASIL DAN PEMBAHASAN

Berikut ini merupakan hasil-hasil pengujian yang telah dilakukan. Pada gambar 6 terlihat grafik perbandingan nilai Avalanche Effect (AE) antara algoritma Sosemanuk dan Dicing. Dari grafik tersebut terlihat bahwa bahwa nilai AE untuk algoritma Sosemanuk lebih besar dari algoritma Dicing sehingga dapat disimpulkan bahwa algoritma Sosemanuk lebih aman dari serangan. Hal ini disebabkan karena jumlah putaran pada algoritma Sosemanuk lebih banyak sehingga *keystream* Sosemanuk lebih acak.

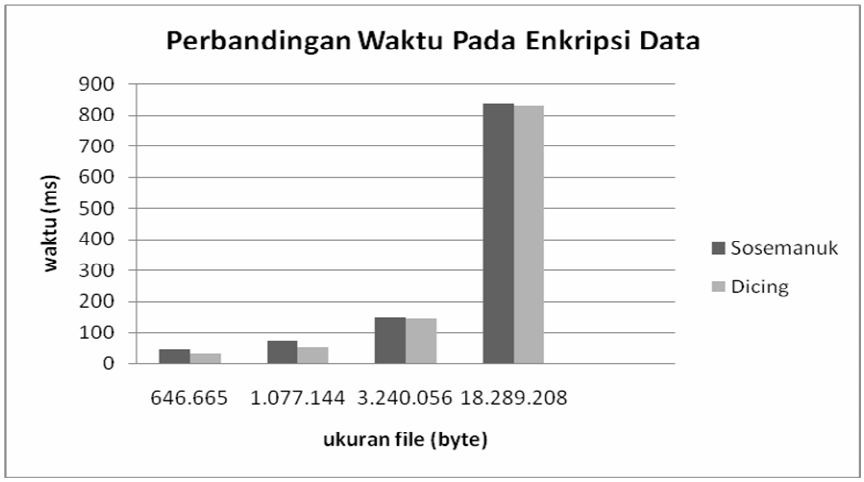


Gambar 6: Grafik Perbandingan Nilai AE Kunci

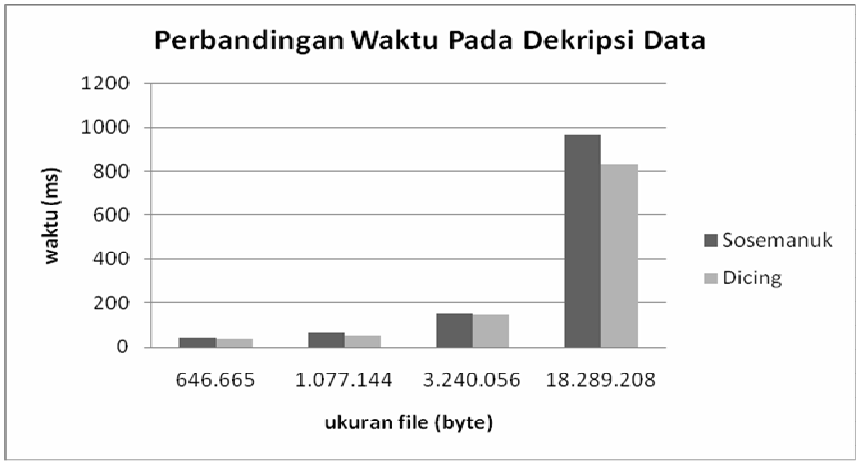
Pada gambar 7 dan 8 disajikan grafik waktu yang diperlukan pada proses enkripsi dan dekripsi antara algoritma Sosemanuk dan Dicing dengan inputan berupa 4 buah file seperti telah disebutkan di atas. Pengujian untuk setiap jenis file masing-masing dilakukan sebanyak 5 kali. Dari gambar 7 terlihat bahwa waktu yang diperlukan untuk proses enkripsi pada algoritma Sosemanuk lebih besar daripada waktu enkripsi pada algoritma Dicing. Hasil tersebut berlaku untuk semua jenis file dengan ukuran berbeda-beda. Rata-rata banyaknya data yang dapat diproses per satuan waktu dihitung dengan persamaan:

$$data\_per\_waktu = \frac{\sum ukuran\_rata2}{\sum waktu\_rata2}$$

Dari hasil pengujian dapat diketahui bahwa rata-rata banyaknya data yang dapat diproses pada saat enkripsi untuk algoritma Sosemanuk sebesar 17,91 kB/milidetik, sedangkan untuk algoritma Dicing sebesar 21,23 kB/milidetik.



Gambar 7: Grafik Perbandingan Waktu Pada Proses Enkripsi



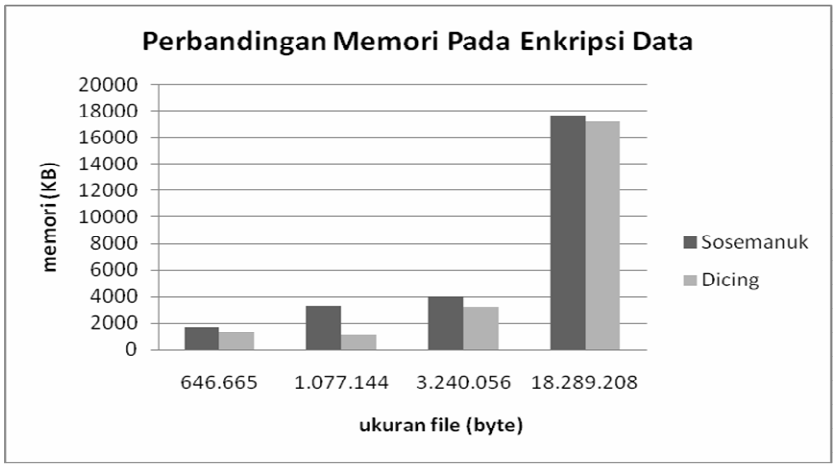
Gambar 8: Grafik Perbandingan Waktu Pada Proses Dekripsi

Dari gambar 8 terlihat bahwa waktu yang diperlukan untuk proses dekripsi pada algoritma Sosemanuk lebih besar daripada waktu enkripsi pada algoritma Dicing. Hasil tersebut berlaku untuk semua jenis file dengan ukuran berbeda-beda. Dari hasil pengujian dapat diketahui bahwa rata-rata banyaknya data yang dapat diproses pada saat dekripsi untuk algoritma Sosemanuk sebesar 18,25 kB/milidetik, sedangkan untuk algoritma Dicing sebesar 20,55 kB/milidetik. Berdasarkan rata-rata banyaknya data yang dapat dienkripsi dan didekripsi per milidetik, maka dapat ditentukan perbedaan kecepatan antara algoritma Sosemanuk dan Dicing sbb:

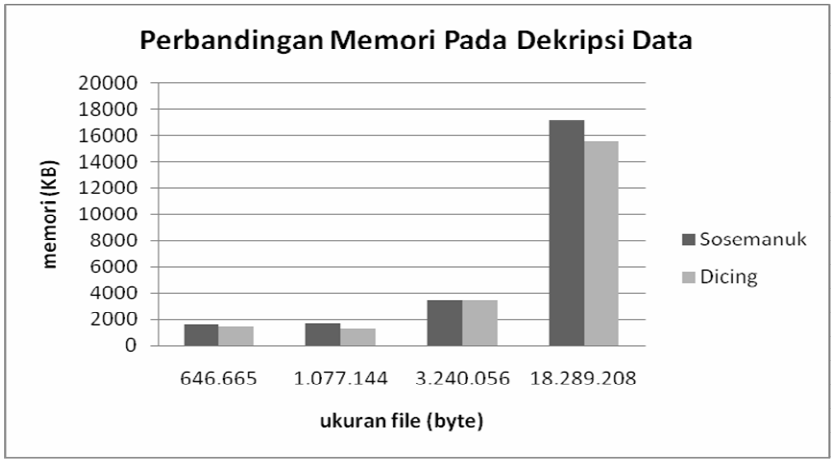
$$\frac{(21{,}23+20{,}55)-(19{,}91+18{,}25)}{(21{,}23+20{,}55)+(17{,}91+18{,}25)} * 100\% = 4{,}77\%$$

Algoritma Dicing 4,77 % lebih cepat daripada algoritma Sosemanuk. Hal ini dikarenakan pada saat pen-*generate*-an kunci, Sosemanuk lebih banyak melakukan putaran dan peng-update-an *state* dibanding Dicing. Lama atau tidaknya proses enkripsi dan dekripsi tidak ditentukan oleh jenis file, tetapi ditentukan oleh ukuran file saja.

Pada gambar 9 dan gambar 10 ditunjukkan perbandingan penggunaan memori pada saat enkripsi dan dekripsi antara algoritma Sosemanuk dan Dicing. Terlihat bahwa untuk proses enkripsi maupun dekripsi algoritma Sosemanuk lebih banyak memerlukan memori daripada algoritma Dicing. Hal ini sejalan dengan lebih kompleksnya proses yang terjadi pada algoritma Sosemanuk.



Gambar 9:Grafik Perbandingan Memori Pada Proses Enkripsi



Gambar 10: Grafik Perbandingan Memori Pada Proses Dekripsi

5. KESIMPULAN

Berdasarkan analisis dari hasil pengujian maka dapat ditarik kesimpulan sbb:

1. Nilai Avalanche Effect (AV) algoritma Sosemanuk lebih besar daripada algoritma Dicing, sehingga algoritma Sosemanuk lebih handal daripada algoritma Dicing.
2. Proses pada algoritma Sosemanuk lebih kompleks daripada algoritma Dicing, sehingga waktu yang diperlukan oleh algoritma Sosemanuk 4,77 % lebih lama dan memori yang diperlukan lebih besar daripada algoritma Dicing.
3. Tipe file tidak berpengaruh terhadap lama waktu enkripsi ataupun dekripsi, karena file dibaca per byte.

Penelitian lanjutan dapat dilakukan agar dapat dihasilkan *ciphertext* yang lebih acak dengan cara melekukan permutasi pada saat proses enkripsi, sehingga *ciphertext* yang dihasilkan dapat lebih menjamin keamanan.

6. DAFTAR PUSTAKA

Ahmadi, Hadi dkk, 2007, *Improved Guess and Determine Attack on Sosemanuk*, http://www.ecrypt.eu.org/stream/sosemanuk.html.

An-ping, Li, 2007, *A New Stream Cipher: DICING*, http://www.ecrypt.eu.org/stream/dicing.html

Answers, 2007, Sosemanuk, http://www.answers. com/library/Wikipedia-cid-1935053910/.

Berbain, C dkk, 2007. *Sosemanuk, a Fast Software Oriented Stream Cipher*, http://www.ecrypt.eu. org/stream/sosemanuk.html.

Ecrypt, 2007, *Dicing*, http://www.ecrypt.eu.org/ stream/dicing.htm/.

Ecrypt, 2007, *Sosemanuk*, http://www.ecrypt.eu. org/stream/sosemanuk.htm/.
Heryanto, Imam, 2006, Pemrograman Borland C++ Builder, Bandung : Penerbit Informatika.
Kurniawan, Yusuf, 2004, *Kriptografi Keamanan Internet dan Jaringan Komunikasi*, Bandung: Penerbit Informatika.
Munir, Renaldi, 2006, *Kriptografi*, Bandung: Penerbit Informatika.
Pirates, Giles, *Practical Attacks on One Version of Dicing*, http://www. ecrypt.eu.org/stream/ papersdir/051.
Purser, Michael, 1996, *Secure Data Networking*, Norwood : Artech House, Inc.
Schneier, Bruce, 1996, *Apllied Cryptography 2nd Edition*, USA : John Wiley & Sons, Inc.
Tsunoo, Yukiyasu dkk, 2007, *Evaluation of Sosemanuk with Regard to Guess and Determine Attacks*, http://www.ecrypt.eu.org/stream/sosemanuk .html.
Wikipedia, 2007, *Cryptography*, http://eu.wikipe dia.org/wiki/cryptography.
Wikipedia, 2007, *Dicing*, http://eu.wikipedia.org /wiki/dicing.
Wikipedia, 2007, *Sosemanuk*, http://eu.wikipedia. org/wiki/sosemanuk.
Wikipedia, 2008, Avalanche Effect, http://en. wikipedia.org/wiki/Avalanche-effect.