ABSTRAK

Nama : **Irham Hafizi**

Program Studi : **Teknik Informatika**

Judul : **Perbandingan Kinerja *Self-Generated True Random Number Generator* Dengan *Pseudorandom Number Generator* Dalam Kriptografi Audio**

Dosen Pembimbing : **Husni, Msc.**

Munculnya pandemi virus *corona* mengakibatkan perubahan cara berkomunikasi yang sebelumnya dilakukan secara tatap muka menjadi komunikasi dalam jaringan internet atau *online*. Komunikasi dalam jaringan memiliki beberapa ancaman, salah satunya adanya pihak yang tidak dikenal dapat melihat informasi yang dikirim maupun diterima. Kriptografi dapat digunakan sebagai solusi dalam meningkatkan keamanan informasi dalam berkomunikasi dalam jaringan. Kriptografi adalah ilmu yang menangani bagian desain algoritma pada proses enkripsi dan dekripsi. Pada proses enkripsi dan dekripsi dibutuhkan kunci rahasia yang sulit ditebak, sehingga pihak tidak dikenal tidak dapat melihat informasi yang sebenarnya. Urutan bilangan acak dapat digunakan sebagai kunci, karena sulit ditebak. Terdapat dua jenis penghasil bilangan acak, yaitu *true random number generator* dan *pseudorandom number generator*. Tugas akhir ini membahas perbandingan kinerja kedua jenis penghasil bilangan acak dalam menghasilkan urutan bilangan acak yang dapat digunakan dalam kriptografi berkas audio. Algoritma *pseudorandom number generator* yang digunakan adalah *blum blum shub* yang menggunakan perhitungan matematis dalam menghasilkan bilangan acak, sedangkan algoritma *true random number generator* yang digunakan adalah *self-generated true random number generator* yang menggunakan berkas yang dienkripsi sebagai sumber entropi dalam menghasilkan urutan bilangan acak. Algoritma XOR *cipher* digunakan dalam proses enkripsi dan dekripsi berkas audio. Kedua berkas kunci yang ebrisi urutan bilangan acak yang dihasilkan oleh kedua jenis penghasil bilangan acak diuji dengan pengujian statistik *frequency (monobit)* yang menghasilkan bahwa penghasil bilangan acak dengan metode *pseudorandom number generator* algoritma *blum blum shub* dapat menghasilkan urutan bilangan acak yang lebih baik tingkat acaknya dibandingkan dengan *self-generated true random number generator*.

Kata kunci: kriptografi, penghasil bilangan acak, *self-generated*

ABSTRACT

The emergence of the corona virus pandemic has resulted in changes in the way of communicating which was previously done face-to-face to communication on the internet or online. Communication in the network has several threats, one of which is that unknown parties can see the information to be sent or received. Cryptography can be used as a solution to improve information security in communicating in the network. Cryptography is a science that deals with the design of algorithms for encryption and decryption. The encryption and decryption process requires a secret key that is difficult to guess, so that unknown parties cannot see the actual information. Sequences of random numbers can be used as keys, as they are difficult to guess. There are two types of random number generators, namely true random number generators and pseudorandom number generators. This final project discusses the comparison of the performance of the two types of random number generators in generating a sequence of random numbers that can be used in audio file cryptography. The pseudorandom number generator algorithm used is blum blum shub which uses mathematical calculations to generate random numbers, while the true random number generator algorithm used is a self-generated true random number generator that uses an encrypted file as a source of entropy to generate a sequence of random numbers. XOR cipher algorithm is used in the encryption and decryption of audio files. The two key files which generate random number sequences generated by both types of random number generators are tested by frequency (monobit) statistical testing which results that random number generators using the pseudorandom number generator method blum blum shub algorithm can produce random number sequences with a better random rate than with self-generated true random number generator.

Keywords: cryptography, random number generator, self-generated