Implementasi Kriptografi Transmisi Teks Menggunakan Mikrokontroler

Adharul Muttaqin, Zainul Abidin

***Abstrak* – Keamanan data pengiriman teks dengan menggunakan mikrokontoler dapat diwujudkan dengan mengimplementasikan teknik enkripsi. Akan tetapi memori mikrokontroler yang terbatas dapat menjadi penghambat karena beberapa teknik kriptografi memerlukan memori yang cukup besar untuk ukuran mikrokontroler. Penelitian ini memberikan contoh teknik kriptografi yang terdiri dari proses enkripsi dan dekripsi dengan memanfaatkan mikrokontroler.**

Hasil pengujian menunjukkan bahwa penggunaan teknik kriptografi tidak memberikan kesalahan pada proses pengiriman data teks delapan karakter dengan rata-rata kesalahan adalah 0%.

Kata Kunci: Mikrokontroler, kriptografi, komunikasi serial, transmisi teks

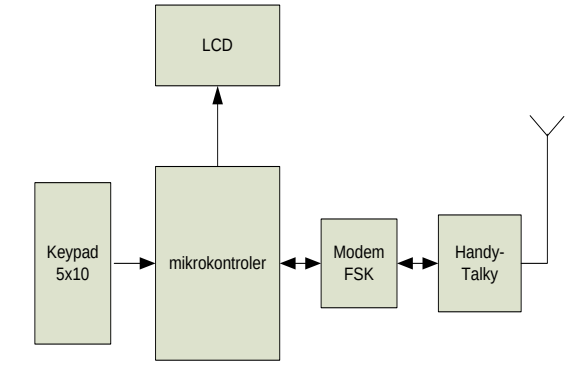
I. PENDAHULUAN

Kriptografi adalah ilmu yang mempelajari bagaimana menjaga keamanan suatu pesan (*plaintext*). Tugas utama kriptografi adalah untuk menjaga agar pesan tetap terjaga kerahasiaannya dari penyadap (*attacker*). Penyadap pesan diasumsikan mempunyai akses yang lengkap dalam saluran komunikasi antara pengirim pesan dan penerima pesan. Penyadapan sering terjadi pada komunikasi melalui saluran internet maupun saluran telepon.

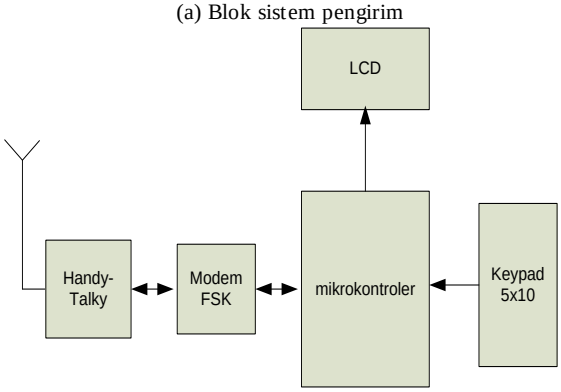
Pada komunikasi serial yang memanfaatkan mikrokontorler, sering dihadapkan pada masalah pemilihan teknik enkripsi yang kuat dan tidak mudah dipecahkan. Penggunaan mikrokontroler dihadapkan pada keterbatasan memori pemroses dan fungsi matematis yang dimiliki. Penelitian inimencoba memberikan solusi tentang bagaimana penerapan teknik kriptografi dengan menggunakan mikrokontroler.

II. IMPLEMENTASI TRANSMISI TEKS

Gambar 1 menunjukkan contoh sistem transmisi teks yang terdiri dari bagian pemgirim dan penerima. Untuk komunikasi terbuka seperti ini, diperlukan proses kriptografi atau enkripsi/ dekripsi agar untuk mencegah informasi sampai kepada yang tidak berhak. Pengirim mengetikkan kunci melalui *keypad*, *keypad* tersebut akan mengubah data teks kemudian pada mikrokontroler dilakukan proses ekspansi kunci dan menampilkan kunci maupun hasil ekspansi kunci ke LCD. Selanjutnya teks yang dienkripsi dan teks asli ditampilkan pada LCD.



(a) Blok sistem pengirim



(b) Blok sistem penerima

Gambar 1 Diagram Blok Perancangan Sistem

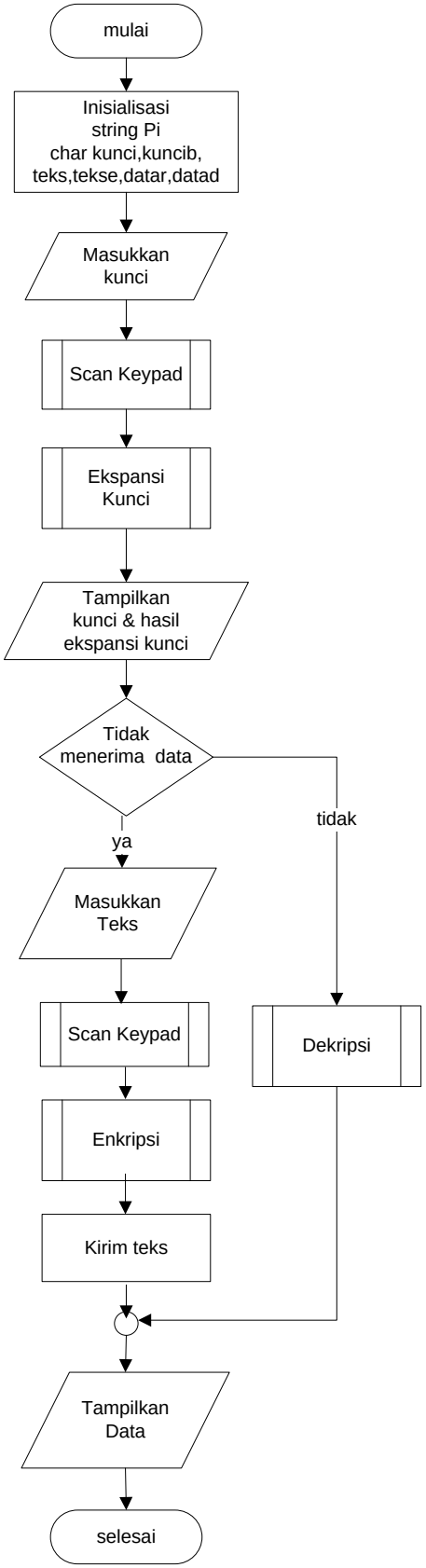
Untuk transmisi, hanya teks terenkripsi yang akan dikirimkan. Pada sisi penerima, teks kemudian didekripsi berdasarkan kunci yang disepakati untuk memperoleh data aslinya. Dengan demikian teks/ pesan yang dikirimkan tidak akan bisa diterima oleh *user* lainnya yang tidak mengetahui kunci dan mempunyai teknik dekripsi yang sesuai.

A. Program Utama

Program utama merupakan program yang pertama kali dijalankan oleh mikrokontroler dan memanggil atau menjalankan subprogram yang lain. Perancangan program utama tersebut sesuai dengan ***flowchart*** dalam

Adharul Muttaqin adalah dosen jurusan Teknik `Elektro Universitas Brawijaya Malang. Penulis dapat dikontak di Jurusan Teknik Elektro Universitas Brawijaya Malang, Jl. MT. Haryono 167 Malang. Telp 0341554166. email adharul@brawijaya.ac.id
Zainul Abidin adalah mahasiswa program sarjanaTeknik Elektro Universitas Brawijaya.

Gambar 2.



Gambar 2 Flowchart Program Utama

Flowchart program utama di atas mengacu pada proses pengiriman atau penerimaan pesan oleh *user* yang dijelaskan pada perancangan sistem. Perancangan program untuk kriptografi mengacu pada algoritma blowfish yang diambil pada beberapa bagiannya saja. Hal tersebut dikarenakan kebutuhan memori untuk algoritma blowfish tidak dapat dipenuhi oleh mikrokontroler.

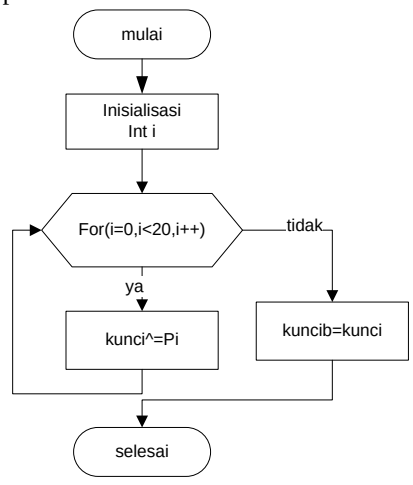
B. Sub Program Ekspansi Kunci

Ekspansi kunci ini dilakukan untuk mengacak kunci yang sebenarnya menjadi kunci yang baru. Kunci tersebut harus dikomputasikan pada saat awal, sebelum pengkomputasian enkripsi dan dekripsi data. Metode pengacakan kunci ini sesuai dengan *flowchart* pada Gambar 3 dengan penjelasan sebagai berikut:

1. Dilakukan inisialisasi Pi dengan string yang tetap. String ini terdiri dari 16 digit heksadesimal. Berikut ini adalah 20 string Pi:

P_1 =0x243f6a882ffd72db	P_{11}=0xbe5466cf636920d8
P_2 =0x85a308d3d01adfb7	P_{12}=0x34e90c6c71574e69
P_3 =0x13198a2eb8e1afed	P_{13}=0xc0ac29b7a458fea3
P_4 =0x037073446a267e96	P_{14}=0xc97c50ddf4933d7e
P_5 =0xa4093822ba7c9045	P_{15}=0x3f84d5b50d95748f
P_6 =0x299f31d0f12c7f99	P_{16}=0xb5470917728eb658
P_7 =0x082efa9824a19947	P_{17}=0x9216d5d9718bcd58
P_8 =0xec4e6c89 b3916cf7	P_{18}=0x8979fb1b82154aee
P_9 =0x452821e60801f2e2	P_{19}=0xd1310ba67b54a41d
P_{10}=0x38d01377858efc16	P_{20}=0x98dfb5acc25a59b5

2. Kunci yang dimasukkan oleh *user* dalam bentuk teks 8 karakter diubah ke dalam bentuk heksadesimal.
3. Dilakukan operasi XOR P_1 dengan 64 bit kunci, XOR P_2 dengan 64 bit hasil XOR pertama kunci, XOR P_3 dengan 64 bit hasil XOR kedua kunci dan seterusnya untuk setiap bit dari Pi (sampai P_{20}) sehingga didapatkan string 64 bit sebagai kunci baru.
4. Kunci baru tersebut disimpan sementara ke dalam "`kuncib`" dan *user* bisa melakukan enkripsi atau dekripsi.



Gambar 3 Flowchart Program Ekspansi Kunci

C. Sub Program Enkripsi

Enkripsi data ini dilakukan untuk mengacak data dengan melibatkan kunci baru. Metode enkripsi ini terdiri dari iterasi fungsi sederhana sebanyak 20 kali. Setiap

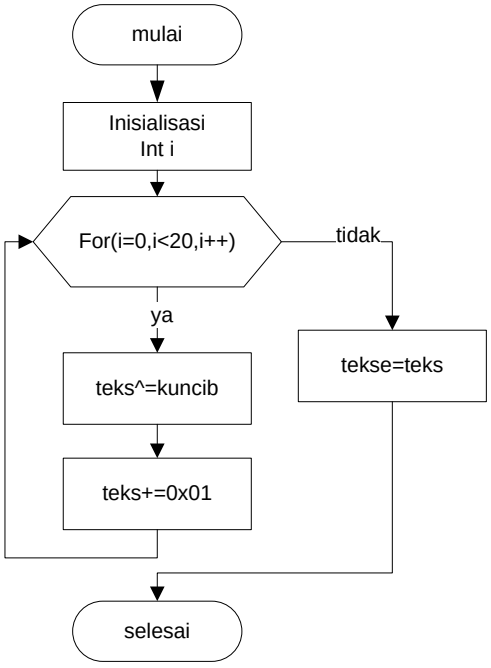
putaran terdiri dari operasi XOR dan penambahan 64 bit. Metode ini sesuai dengan *flowchart* pada Gambar 4 dengan penjelasan sebagai berikut:

Delapan karakter teks yang akan dienkripsi, diuabah ke dalam bentuk heksadesimal.

Dilakukan operasi XOR heksadesimal 8 karakter tersebut dengan kunci baru kemudian hasilnya ditambahkan dengan 0x0101010101010101.

Berikutnya dilakukan operasi XOR hasil perhitungan di atas dengan kunci baru kemudian tambahkan hasilnya dengan 0x0101010101010101 lagi dan seterusnya sampai pada putaran ke-20.

Setelah itu akan didapatkan 64 bit *chipertext* yang telah dienkripsi dan siap ditransmisikan.

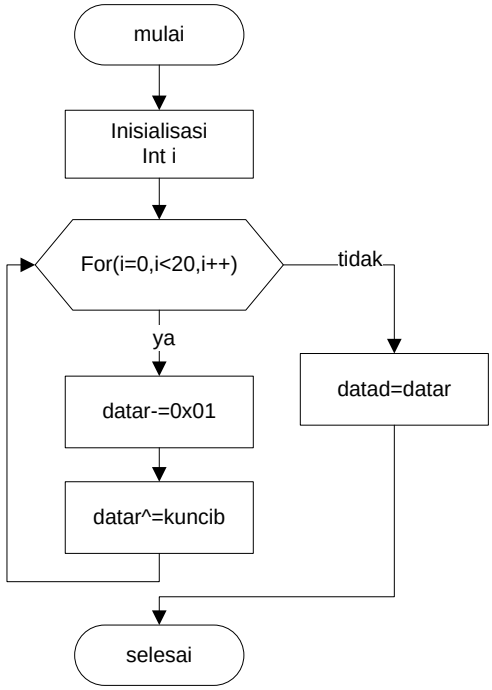


Gambar 4 Flowchart Program Enkripsi

D. Sub Program Dekripsi

Dekripsi tidak berbeda dengan enkripsi, yaitu dengan membalik prosesnya dengan menggunakan operasi XOR dan pengurangan. Proses dalam metode dekripsi ini sesuai dengan *flowchart* pada Gambar 5 dengan penjelasan sebagai berikut:

1. Data *chipertext* 64 bit yang telah diterima, diubah mikrokontroler dalam heksadesimal dan dikurangi dengan 0x0101010101010101.
2. Dilakukan operasi XOR hasil pengurangan dengan kunci baru.
3. Proses di atas sampai 20 putaran sehingga dihasilkan 64 bit data terdekripsi.
4. Setelah itu 64 bit data tersebut diubah ke dalam teks (*plaintext*) lagi dan siap dibaca *user*.



Gambar 5 Flowchart Program Dekripsi

E. Format Data

Format data yang digunakan dalam komunikasi teks dengan menggunakan kriptografi ini harus dirancang seunik mungkin. Hal ini untuk mempermudah penerimaan dan pembacaan teks/pesan yang dikirimkan. Dalam proses penerimaan data harus terdapat sub proses seleksi kondisi untuk menentukan karakter yang pertama kali harus diterima. Hal tersebut merupakan penanda awal dari sebuah pesan yang dikirimkan. Untuk menghindari kesalahan pembacaan pesan, perlu juga diberikan karakter penanda akhir. Format data dalam sistem komunikasi teks ini dapat dilihat pada Gambar 6.

Kepala	Teks/pesan	Ekor
Titik (.)	xxxxxxxxx	@

Gambar 6 Format Data

Penjelasan dari format data adalah sebagai berikut:

Kepala : sebuah karakter '.' yang ditempatkan di bagian awal yang menandakan karakter pesan sebenarnya maupun pesan hasil proses enkripsi dimulai setelah karakter tersebut

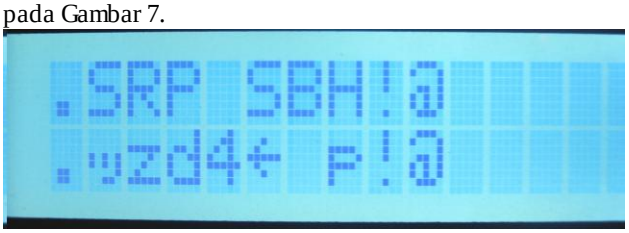
Teks/pesan : merupakan kumpulan 8 karakter (16 digit heksadesimal)

Ekor : sebuah karakter '@' yang ditempatkan di bagian akhir yang menandakan karakter pesan sebenarnya maupun pesan hasil proses enkripsi/dekripsi diakhiri sebelum karakter tersebut.

III. PENGUJIAN IMPLEMENTASI KRIPTOGRAFI

A. *Data Pengujian*

Data hasil pengujian enkripsi didapatkan dengan mengamati hasil proses enkripsi pada sisi pengirim dan hasil proses dekripsi pada sisi penerima. Dengan sistem komunikasi yang berjalan dengan baik, pengiriman pesan dapat diterima dengan baik pula. Dalam pengujian ini dilakukan pengiriman pesan teks ”S R P S B H !” dengan mematuhi aturan format data seperti dijelaskan pada perancangan format data (Gambar 4. 11). Pesan sesungguhnya dan pesan yang sudah dienkripsi dapat dilihat pada alah satu tampilan LCD pengirim seperti pada Gambar 7.



Gambar 7 Tampilan LCD Setelah Proses Enkripsi

B. *Data Pengujian pada Sisi Penerima dengan Kunci yang Benar*

Data pengujian tersebut berupa tampilan LCD yang berisi pesan terenkripsi dan hasil dekripsi seperti terlihat pada Gambar 8. Dengan membandingkan tampilan LCD setelah proses enkripsi dan proses dekripsi dapat disimpulkan bahwa setiap karakter dapat diterima dengan benar.

Gambar 8 Tampilan LCD Setelah Proses Dekripsi dengan Kunci yang Benar

TABEL 1 TABEL PENGUJIAN PENERAPAN ENKRIPSI/DEKRIPSI UNTUK TEKS 8 KARAKTER

Data ke	Jumlah karakter benar	% Kesalahan
1	8	0
2	8	0
3	8	0
4	8	0
5	8	0
6	8	0
7	8	0
8	8	0
9	8	0
10	8	0

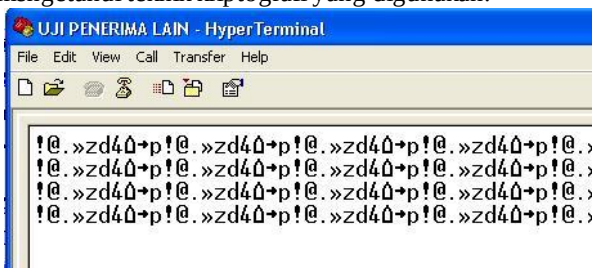
C. *Data Pengujian pada Sisi Penerima dengan Kunci yang Salah*

Data pengujian tersebut berupa tampilan LCD yang berisi pesan terenkripsi dan hasil dekripsi seperti terlihat pada Gambar 9. Dengan membandingkan tampilan LCD setelah proses enkripsi dan proses dekripsi dapat disimpulkan bahwa karakter hasil proses dekripsi tidak sesuai, sehingga *user* tidak bisa mendapatkan informasi yang sebenarnya. Hal ini memastikan bahwa informasi dinyatakan aman dari *user* yang tidak mengetahui kunci yang benar.

Gambar 9 Tampilan LCD Setelah Proses Dekripsi dengan Kunci yang Salah

D. *Data Pengujian dengan Alat Penerima yang Berbeda*

Data pengujian tersebut berupa tampilan *hyperterminal* PC dari alat penerima. Tampilan pada Gambar 9 dan 10 tersebut menunjukkan bahwa alat penerima ini hanya bisa mendapatkan *chipertext* tetapi tidak bisa mendapatkan informasi yang sebenarnya (*plaintext*). Hal ini memastikan bahwa informasi dinyatakan aman dari penyadap (*attacker*) yang tidak mengetahui teknik kriptografi yang digunakan.



Gambar 10 Tampilan *Hyperterminal* PC pada Alat Penerima yang Berbeda

IV. KESIMPULAN

Berdasarkan pembahasan tentang sistem komunikasi teks menggunakan *handy talky* dengan menerapkan kriptografi sebagai keamanan yang telah dipaparkan di atas, dapat diperoleh kesimpulan sebagai berikut :

1. Penerapan teknik kriptografi dengan mikrokontroler

yang mungkin diterapkan adalah dengan menggunakan pengacakan kode heksadesimal dari setiap karakter dengan menggunakan operasi logika XOR, operasi penambahan dan operasi pengurangan.

2. Penerapan kriptografi yang dilakukan adalah dengan mengekspansi karakter kunci yang telah dimasukkan dengan melibatkan operasi logika XOR, melakukan enkripsi pada teks yang telah dimasukkan dengan melibatkan operasi logika XOR dan penambahan pada saat akan mengirimkan pesan, dan melakukan dekripsi dengan melibatkan operasi XOR dan operasi pengurangan pada saat akan menampilkan pesan yang diterima.
3. Performansi penerapan kriptografi ini ditunjukkan dengan nilai rata-rata persentase kesalahan sebesar 0% yang menunjukkan bahwa proses enkripsi dan dekripsi berlangsung sempurna.

DAFTAR PUSTAKA

.

[1] Ariyus, Dony. 2006. *Kriptografi Keamanan Data dan Komunikasi*. Penerbit Graha Ilmu Yogyakarta

[2] Atmel, 1997, *Flash Microcontroller: Architectural Overview*, Atmel Inc. (**http://www.atmel.com**), USA.

[3] Bagus, Triadi. 2006. *Kriptografi* http://triadi.bagus.googlepages.com/Enkripsi-Dekripsi.pdf. diakses tanggal 14 Desember 2007.

[4] Cooper, William D. 1985. Instrumentasi Elektronika dan Teknik Pengukuran. Edisi Kedua. Jakarta Pusat. Penerbit : Erlangga.

[5] Kurniawan, Yusuf. 2004. *Kriptografi Keamanan Internet dan Jaringan Komunikasi*. Penerbit Informatika Bandung.

[6] Ratih. 2007. *Studi dan Implementasi Algoritma Blowfish untuk Aplikasi dan Dekripsi file*. http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-077.pdf. diakses tanggal 7 Februari 2008.

[7] Schneier, Bruce. 1995. *Applied Cryptography-Protocols, Algorithms, and Source Code in C*, 2nd. New York. John Wiley & Son.

[8] Sukiswo.2005. *Perancangan Telemetri Suhu dengan Modulasi Digital FSK-FM* http://www.elektro.undip.ac.id/transmisi/des05/sukiswodes05.PDF. Diakses tanggal 7 Februari 2008.

[9] Wardhana, Lingga. 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATMega8535*. Yogyakarta : Andi Offset.