Perangkat Lunak Pembelajaran Protokol Secret Sharing Dengan Algoritma Asmuth – Bloom

Marto Sihombing [1)], Erich Gunawan [2)]
STMIK IBBI
Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548
E-mail : martosihombing@yahoo.com

Abstrak

Algoritma kriptografi konvensional tidak dapat diterapkan untuk memecahkan sebuah *plaintext* (pesan) menjadi beberapa buah *ciphertext* karena algoritma kriptografi konvensional hanya dapat menghasilkan sebuah *ciphertext* dari sebuah *plaintext* (pesan). Dalam kasus ini, maka dapat diterapkan protokol kriptografi, yaitu protokol *secret sharing Asmuth-Bloom*. Proses kerja dari protokol *secret sharing Asmuth-Bloom* ini terbagi menjadi dua bagian yaitu proses pembentukan *shadow* dan proses penggabungan *shadow*. Proses pembentukan *shadow* akan dimulai dari proses pengisian pesan dan kumpulan nilai d beserta nilai m, n dan bilangan prima p. Setelah itu, proses dilanjutkan dengan pengubahan pesan ke bentuk ASCII *Code*. Sementara itu, proses penggabungan *shadow* akan dimulai dari pengisian m buah *share*. Setelah itu, proses dilanjutkan dengan perhitungan setiap karakter dengan menggunakan *Chinese Remainder Theorem*. Proses akan diakhiri dengan penggabungan setiap karakter yang dihasilkan sehingga diperoleh pesan semula. Prosedur kerjanya mencakup proses pembentukan kunci, proses pembentukan *shadow* dan proses penggabungan *shadow*. Perangkat lunak juga menyediakan antarmuka untuk melakukan proses pemecahan *file* teks menjadi n buah *file shadow* dan penggabungan m buah *file shadow* menjadi *file* teks semula. Selain itu, perangkat lunak juga menyediakan teori-teori dasar yang berhubungan dengan kedua protokol kriptografi dan laporan hasil proses perhitungan disimpan ke dalam *text file* berekstensi *.txt.

Kata kunci: Kriptografi, *plaintext, ciphertext, secret sharing Asmuth-Bloom*.

Abstract

Algorithm cryptography conventional cannot be applied solve a plaintext become some ciphertext because algorithm cryptography conventional can only ciphertext from a plaintext. In this case, hence can be applied by protocol cryptography, that protocol of secret Asmuth-Bloom sharing. Process work from protocol secret this Asmuth-Bloom sharing divided to become two part of that process forming process and shadow merger shadow. Process forming shadow will start from process admission filling and message assess d along with value of m, prime number and n p. Afterwards, process to continued with distorting message to form of ASCII Code. Meanwhile, process merger of shadow will start from admission filling share. Afterwards, process to be continued with calculation every character by using Chinese Remainder Theorem. Process will terminate with merger every character is so that obtained by message initialy. Its working procedure include process forming of key, process forming process and shadow merger of shadow. Software also provide interface to process resolving of text file become file n shadow and merger of file m shadow become text file initialy. Besides, software also provide elementary theorys related to both protocol cryptography report and result calculation process kept into file text extention *. txt.

Keywords*: Cryptography*, *plaintext, ciphertext, secret sharing Asmuth-Bloom*.

1. Pendahuluan

Algoritma kriptografi konvensional dapat digunakan untuk mengamankan data dalam proses komunikasi. Algoritma kriptografi konvensional tersebut hanya dapat menghasilkan sebuah *ciphertext* dari sebuah *plaintext* (pesan). Protokol *secret sharing* dan *secret splitting* sama-sama dapat digunakan untuk memecahkan sebuah pesan menjadi n buah *ciphertext* berbeda yang dapat dibagikan kepada n orang. Perbedaannya yaitu untuk mendapatkan pesan semula, protokol *secret splitting* memerlukan n buah *ciphertext* tersebut, sedangkan protokol *secret sharing* menerapkan skema (m,n)-threshold yaitu hanya memerlukan m buah *ciphertext* dari total n buah *ciphertext* yang tersedia untuk membentuk pesan kembali, dimana $m \leq n$. Dalam literatur kriptografi, terdapat banyak algoritma yang menerapkan konsep dari protokol *secret sharing* tersebut. Salah satu algoritma dari protokol *secret sharing* tersebut adalah

algoritma *Asmuth-Bloom*. Algoritma ini menggunakan bilangan prima dan bilangan acak untuk meningkatkan keamanannya. Selain itu, algoritma ini juga memerlukan n buah deretan bilangan d_i yang harus memenuhi persyaratan tertentu. Proses pembentukan *ciphertext* dari algoritma *Asmuth-Bloom* ini relatif mudah, yaitu hanya dengan melakukan operasi penjumlahan modulo. Sedangkan, proses pembentukan pesan semula relatif rumit, yaitu memerlukan bantuan teorema *Chinese Remainder*. Proses kerja dari protokol *secret sharing* cukup panjang dan rumit apabila dilakukan perhitungan secara manual, maka perlu dirancang sebuah perangkat lunak pembelajaran yang mampu untuk menampilkan proses kerja dari protokol *secret sharing Asmuth-Bloom* tersebut.

2. Metode Penelitian

Berdasarkan masalah "**Perancangan Perangkat Lunak Pembelajaran Protokol Secret Sharing dengan Algoritma Asmuth - Bloom**". Adapun metode yang digunakan dalam pembuatan PENELITIAN ini adalah :

1. Metode Penelitian, yaitu :

   Studi Kepustakaan Teknik pengumpulan data dengan membaca buku-buku pustaka yang merupakan penunjang dalam memperoleh data untuk melengkapi dalam penyusunan laporan yang berhubungan dengan masalah.
2. Metode Pengembangan Perangkat Lunak

Metode yang digunakan yaitu paradigma *Waterfall (Classic Life Cycle)*,

3. Hasil dan Analisis

3.1 Kriptografi

Kriptografi berasal dari bahasa yunani yang terdiri dari dua kata yaitu kripto dan graphia, dimana kripto berarti secret (rahasia) dan graphia berarti writing (tulisan). Sehingga kriptografi dapat diartikan menjadi ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain.

Berdasarkan jumlah kunci yang digunakan, terdapat dua jenis sistem kriptografi yaitu sistem kriptografi kunci rahasia (*secret-key cryptography*) dan sistem kriptografi kunci publik (*public-key cryptography*).

Kriptografi Kunci Rahasia (*secret-key cryptography*)

Kriptografi Kunci Publik (*public-key cryptography*)

Konsep Matematis Kriptografi

Kunci publik dalam metode kriptografi banyak yang memerlukan bilangan prima. Bilangan prima adalah bilangan *integer* yang lebih besar dari satu yang memiliki faktor bilangan satu dan bilangan itu sendiri. Cara yang salah untuk mendapatkan bilangan prima adalah dengan membangkitkan bilangan acak dan kemudian mencoba memfaktorkannya. Cara yang benar adalah membangkitkan bilangan acak dan kemudian menguji apakah merupakan bilangan prima. Terdapat beberapa metode tes peluang prima, tes menentukan apakah suatu bilangan termasuk bilangan prima atau bukan dengan tingkat keyakinan tertentu. Jadi kita tidak yakin seratus persen bahwa bilangan yang kita tes adalah betul-betul bilangan prima.

Metode Tes Prima Rabin-Miller

Algoritma sederhana yang digunakan oleh semua orang dirancang oleh Michael Rabin dengan berdasarkan beberapa ide dari Gary Miller.

Implementasi Pembangkit Bilangan Prima

Dalam dunia nyata, implementasi pembangkitan bilangan prima dapat berlangsung dengan sangat cepat.

Aritmatika Modular

Aritmatika modular merupakan operasi matematika yang banyak diimplementasikan pada metode kriptografi. Aritmatika modulo mengambil bilangan tak berhingga dan menggulungnya dalam suatu lingkaran terbatas. Semua bilangan yang melintasi point yang sama pada titik lingkaran adalah kongruen.

Kongruen dalam Modulo

Kadang-kadang dua buah bilangan bulat, a dan b, mempunyai sisa yang sama jika dibagi dengan bilangan bulat positif m. Hal itu dapat dikatakan bahwa a dan b kongruen dalam modulo m, dan dilambangkan sebagai :

$$a \equiv b \pmod{m}$$

Jika a tidak kongruen dengan b dalam modulus m, maka ditulis :

$$a \not\equiv b \pmod{m}$$

Definisi formal dari kekongruenan dinyatakan sebagai berikut :

Misalkan a dan b adalah bilangan bulat dan m adalah bilangan > 0, maka $a \equiv b \pmod{m}$ jika m habis membagi a – b.

Kekongruenan $a \equiv b \pmod{m}$ dapat pula dituliskan dalam hubungan :

$$a = b + km$$

yang dalam hal ini sembarang k adalah bilangan bulat.

Berdasarkan definisi aritmetika modulo, maka dapat dituliskan a mod m = r sebagai :

$$a \equiv r \pmod{m}$$

Sifat-sifat pengerjaan hitung pada aritmetika modulo, khususnya terhadap operasi perkalian dan penjumlahan dapat dinyatakan dalam Teorema 2.1 berikut :

Invers Modulo

Di dalam aritmetika bilangan riil, invers dari perkalian adalah pembagian. Misalnya invers dari 4 adalah ¼, karena 4 x ¼ = 1. Di dalam aritmetika modulo, masalah menghitung invers modulo lebih rumit.

Jika a dan m relatif prima dan m > 1, maka dapat ditemukan invers dari a modulo m. Invers dari a modulo m adalah bilangan bulat a^{-1} sedemikian sehingga :

$$a \cdot a^{-1} \equiv 1 \pmod{m}$$

Pembuktian invers modulo ini sangat mudah, seperti terlihat pada penjabaran berikut ini :

GCD(a, m) = 1.

pa + qm = 1, yang mengimplikasikan bahwa $pa + qm \equiv 1 \pmod{m}$.

Karena $qm \equiv 0 \pmod{m}$, maka :

$$pa \equiv 1 \pmod{m}$$

Kekongruenan ini berarti bahwa p adalah invers dari a modulo m.

Kekongruenan Linier

Kekongruenan linier adalah kongruen yang berbentuk :

$$ax \equiv b \pmod{m}$$

dengan m adalah bilangan bulat positif, a dan b adalah sembarang bilangan bulat, dan x adalah peubah. Bentuk kongruen linier berarti menentukan nilai-nilai x yang memenuhi kekongruenan tersebut. Metode yang sederhana untuk mencari nilai-nilai x tersebut adalah sebagai berikut :

$$ax = b + km$$

yang dapat disusun menjadi :

$$x = (b + km) / a$$

dengan k adalah sembarang bilangan bulat. Cobalah nilai-nilai k = 0, 1, 2, … dan k = -1, -2, … ke dalam persamaan yang terakhir untuk menghasilkan x sebagai bilangan bulat.

Metode lain untuk mencari solusi kekongruenan linier adalah dengan menggunakan invers modulo. Caranya serupa dengan pencarian solusi pada persamaan linier biasa, seperti pada :

$$4x = 12$$

Untuk mencari solusi persamaan di atas, kalikan kedua ruas dengan invers perkalian dari 4, yaitu ¼,

¼ . 4x = ¼ . 12

x = 3

Terapkan metode seperti ini pada kekongruenan linier pada :

$$4x \equiv 3 \pmod{9}$$

Kalikan kedua ruas dengan invers dari 4 (mod 9), yang dapat dicari dengan menggunakan algoritma *Extended Euclidean*, dan hasil yang diperoleh adalah – 2.

$$-2 \cdot 4x \equiv -2 \cdot 3 \pmod{9}$$

$$-8x \equiv -6 \pmod{9}$$

Karena $-8 \equiv 1 \pmod{9}$, maka :

$$x \equiv -6 \pmod{9}$$

Chinese Remainder Theorem

Pada abad pertama, seorang matematikawan Tiongkok yang bernama Sun Tzu mengajukan pertanyaan sebagai berikut :

"Tentukan sebuah bilangan bulat yang bila dibagi dengan 5 menyisakan 3, bila dibagi 7 menyisakan 5, dan bila dibagi 11 menyisakan 7".

3.2. Protokol Kriptografi

Suatu protokol adalah serangkaian langkah yang melibatkan dua pihak atau lebih dan dirancang untuk menyelesaikan suatu tugas.

Protokol *Secret Sharing*

Protokol kriptografi lainnya adalah *secret sharing*, yang memungkinkan pendistribusian satu rahasia di antara sekumpulan orang yang saling percaya. Protokol *secret sharing* ini menerapkan ***(m,n)-threshold scheme***, yaitu informasi tentang rahasia adalah didistribusikan sedemikian rupa sehingga sembarang ***m*** dari ***n*** orang ***(m ≤ n)*** memiliki informasi yang cukup untuk menentukan (mengetahui) rahasia tersebut, tetapi sembarang set ***m-1*** orang tidak dapat melakukannya. Dalam sembarang *secret sharing scheme*, terdapat kumpulan orang yang terpilih yang informasi kumulatif mereka cukup untuk memecahkan rahasia.

Threshold Scheme

Variabel yang terdapat dalam (m,n)-*threshold scheme* memiliki fungsinya masing-masing seperti terlihat pada rincian berikut:

- Nilai m berarti jumlah bagian yang diperlukan agar pesan dapat dibaca.
- Nilai n berarti jumlah pecahan / bagian dari pesan.

Selain itu variabel yang terdapat dalam (m,n)-*threshold scheme* harus memenuhi ketentuan berikut:

$$m \leq n$$

Cara kerja dari (m,n)-*threshold scheme* dapat dijabarkan sebagai berikut:

- Pesan dibagi menjadi n buah bagian, yang disebut sebagai bayangan (*shadow*) atau bagian (*share*).
- Bagian-bagian tersebut dibagikan kepada n orang, dengan setiap orang mendapatkan satu bagian yang berbeda-beda satu sama lain.
- Tentukan nilai m sehingga diperlukan m buah bagian pesan agar dapat menyusun kembali pesan yang dirahasiakan tersebut.

3.3. Asmuth-Bloom Algorithm

Algoritma *Asmuth-Bloom* menggunakan aritmatika modulo, bilangan prima dan bilangan acak untuk meningkatkan keamanannya. Selain itu, algoritma ini juga memerlukan bantuan teorema *Chinese Remainder* pada saat penggabungan pesan kembali.

Secara garis besar, algoritma *Asmuth-Bloom* ini dapat dibagi menjadi 3 tahapan proses, yaitu:

a. Proses Pembentukan Kunci
   - Tentukan sebuah bilangan prima p, dimana p lebih besar daripada nilai Kode ASCII Pesan M.
   - Tentukan nilai m dan n, dimana $m \leq n$.
   - Tentukan n buah bilangan yang lebih kecil daripada p, yaitu:

     $$d_1, d_2, d_3, \ldots, d_n$$

     sedemikian sehingga:
     - Deretan nilai d dalam urutan menaik, $d_i < d_{i+1}$.
     - Setiap nilai d_i relatif prima terhadap setiap nilai d_i lainnya.
     - $d_1 * d_2 * \ldots * d_m < p * d_{n-m+2} * d_{n-m+3} * \ldots * d_n$.

b. Proses Pemecahan Pesan
   - Tentukan sebuah bilangan acak r.
   - Hitung nilai M'dengan menggunakan rumusan berikut:

     $$M' = M + rp$$

   - Pecahan pesan (*shadow*)-nya adalah:

     $$k_i = M' \bmod d_i$$

c. Proses Penggabungan Pesan

- Tentukan m buah nilai k_i yang ingin digabungkan:

  Misalnya:

  $k_1 = M' \bmod d_1$

  $k_2 = M' \bmod d_2$

  …

  $k_m = M' \bmod d_m$

  dimana nilai k_i dan d_i diketahui.
- Untuk mencari nilai M' digunakan bantuan teorema *Chinese Remainder*.

3.4. Algoritma

Algoritma yang digunakan untuk merancang perangkat lunak pemahaman dan aplikasi algoritma *Secret Sharing Asmuth-Bloom* ini dapat dibagi menjadi :

Algoritma Pembentukan Kunci

Algoritma ini berfungsi untuk menghasilkan nilai-nilai yang akan digunakan pada proses pembuatan *shadow* dan proses penggabungan *shadow*. Nilai-nilai *output* dari algoritma ini, yaitu:

1. Nilai m dan n.
2. Bilangan prima p.
3. Deretan nilai d(1) … d(n).

Algoritma Pembuatan Shadow

Algoritma ini berfungsi untuk menghasilkan pecahan pesan (*shadow*) dari pesan input. Nilai-nilai yang diperlukan oleh algoritma ini, yaitu:

1. Nilai n, bilangan prima p dan deretan bilangan d(1) ... d(n) yang dihasilkan dari proses pembentukan kunci.
2. Pesan *input* dan bilangan acak r.

Sedangkan, *output* dari algoritma ini adalah n buah pecahan pesan (*shadow*) dari pesan *input*. Prosedur kerja dari algoritma pembuatan *shadow* ini dapat dirincikan sebagai berikut:

1. Input pesan.
2. Jika input manual, maka
   a. Input bilangan acak r.
   b. Jika input sama dengan 0 maka kembali ke langkah (a).
3. Jika tidak, maka ambil sebuah bilangan acak r.
4. Untuk i = 1 sampai [panjang pesan], lakukan proses berikut:
   a. Konversikan karakter ke-i dari pesan ke bentuk ASCII Code dan simpan ke variabel M(i).
   b. Hitung nilai M(i)' = M(i) + rp.
5. Untuk j = 1 sampai n, lakukan proses berikut:
   Untuk i = 1 sampai [panjang pesan], lakukan proses berikut:
   Hitung nilai k(j, i) = M(i) mod d(j)

Algoritma Penggabungan Shadow

Algoritma ini berfungsi untuk menghasilkan pecahan pesan (*shadow*) dari pesan input. Nilai-nilai yang diperlukan oleh algoritma ini.

Algoritma Chinese Remainder

Algoritma ini memiliki *input data* yaitu :

1. Variabel *array* dua dimensi nArrNilai, dengan perincian dimensi pertama bernilai sebesar jumlah persamaan dan dimensi kedua bernilai sebesar 2 yaitu nilai 1 untuk bilangan sisa modulo dan nilai 2 untuk bilangan modulo.
2. Variabel nJlh merupakan jumlah persamaan.

Sedangkan, *output* dari algoritma ini berupa nilai invers modular yang memenuhi kedua persamaan modular.

Selain itu, algoritma *Chinese Remainder* ini juga membutuhkan algoritma *Extended Euclidean* untuk menghitung nilai invers modular dari sebuah persamaan modular.

Algoritma Extended Euclidean

Algoritma ini memiliki sebuah *input data* yaitu :

1. Variabel pnValueA11 yaitu bilangan sisa modulo.

2. Variabel pnValueE yaitu bilangan modulo.

Sedangkan, *output* dari algoritma ini berupa nilai invers dari persamaan modular tersebut.

Proses Pembentukan Kunci

Seperti pada algoritma kriptografi kunci publik, proses pembentukan kunci dari algoritma *secret sharing* ini menghasilkan kunci privat dan kunci publik yang akan digunakan dalam proses pembentukan dan penggabungan *shadow*.

Kunci privat dan publik yang terdapat pada algoritma *Secret Sharing Asmuth-Bloom* ini dapat dirincikan sebagai berikut:

1. Kunci publik (*public key*) dari semua *user*, yaitu bilangan prima p.
   Bilangan prima p ini dapat dibangkitkan dengan menggunakan algoritma pembangkitan bilangan prima dari metode Rabin-Miller ataupun di-*input* secara manual dan dites dengan menggunakan algoritma pengujian bilangan prima dari metode Rabin-Miller. Bilangan prima p ini harus lebih besar dari *ASCII Code* pesan. Karena nilai *ASCII Code* terbesar adalah 255, maka nilai bilangan prima p harus lebih besar daripada 255.
2. Kunci privat (*private key*) dari masing-masing *user*, yaitu deretan nilai $d_1 \ldots d_n$.
   Deretan nilai d ini dapat ditentukan secara manual ataupun dihasilkan secara acak dengan memenuhi beberapa persyaratan berikut:
   - Deretan nilai d dalam urutan menaik, $d_i < d_{i+1}$.
   - Setiap nilai d_i relatif prima terhadap setiap nilai d_i lainnya.
   - $d_1 * d_2 * \ldots * d_m < p * d_{n-m+2} * d_{n-m+3} * \ldots * d_n$.

   Selain itu, proses pembentukan kunci juga akan menghasilkan nilai m dan n dimana nilai m merupakan jumlah *shadow* yang diperlukan untuk membentuk pesan dan nilai n merupakan jumlah *shadow* yang diinginkan. Proses pembentukan kunci ini dapat digambarkan dalam bentuk *flowchart*

Proses pembentukan kunci dari algoritma *Secret Sharing Asmuth-Bloom* dimulai dengan menentukan sebuah bilangan prima p, dimana p lebih besar daripada nilai Kode ASCII Pesan M. Setelah itu, dilakukan pengecekan apakah bilangan yang di-input merupakan bilangan prima atau bukan. Jika ya, maka proses dilanjutkan. Jika tidak, maka proses diulangi lagi dari penginputan bilangan prima.

Setelah itu, maka proses dilanjutkan dengan menentukan nilai m dan n, dimana $m \leq n$. Jika nilai m dan n tidak memenuhi ketentuan maka proses diulangi dari proses penginputan nilai m dan n.

Jika nilai m dan n memenuhi ketentuan, maka proses dilanjutkan dengan menentukan n buah bilangan yang lebih kecil daripada p, yaitu: $d_1, d_2, d_3, \ldots, d_n$ sedemikian sehingga:

- Deretan nilai d dalam urutan menaik, $d_i < d_{i+1}$.
- Setiap nilai d_i relatif prima terhadap setiap nilai d_i lainnya.
- $d_1 * d_2 * \ldots * d_m < p * d_{n-m+2} * d_{n-m+3} * \ldots * d_n$.

Jika memenuhi ketentuan, maka proses selesai. Jika tidak, maka proses diulangi lagi dari proses penginputan deretan nilai d.

Proses Pembentukan *Shadow*

Proses pembentukan *shadow* dari algoritma *secret sharing* ini menggunakan *output* dari proses pembentukan kunci yaitu kunci privat dan kunci publik *user*. Proses pembentukan *shadow* dari algoritma *secret sharing* ini dilakukan oleh pembuat pesan. Hasil dari proses ini adalah n buah *shadow* yang akan dibagikan kepada n orang, dimana setiap *shadow* memiliki nilai yang berbeda-beda. Proses pembentukan *shadow* dari algoritma *secret sharing* ini dapat digambarkan dalam bentuk *flowchart*

Proses pembentukan *shadow* dimulai dari proses penginputan pesan dan pengkonversian setiap karakter pesan ke bentuk ASCII Code. Setelah itu, proses dilanjutkan dengan proses penentuan sebuah bilangan acak r. Kemudian, menghitung nilai M'dengan menggunakan rumusan: $M' = M + rp$. Setelah itu, proses diakhiri dengan proses pembentukan *shadow* untuk setiap karakter pesan dengan menggunakan rumusan : $k_i = M' \bmod d_i$.

Proses Penggabungan *Shadow*

Proses penggabungan *shadow* dari algoritma *secret sharing* ini menggunakan *output* dari proses pembentukan kunci yaitu kunci privat dan kunci publik *user*, serta m buah *shadow*. Proses penggabungan *shadow* dari algoritma *secret sharing* ini dilakukan oleh m orang yang ingin mendapatkan pesan semula. Hasil dari proses ini adalah pesan semula yang disembunyikan oleh pembuat pesan.

Proses penggabungan *shadow* menggunakan bantuan teorema *Chinese Remainder* untuk mencari solusi dari sistem kongruen linier yang dibentuk dari gabungan m buah *shadow* dan m buah nilai d_i.

Proses penggabungan *shadow* dari algoritma *Secret Sharing Asmuth-Bloom* ini dapat digambarkan dalam bentuk *flowchart*

Proses penggabungan *shadow* dimulai dari penginputan nilai m buah *shadow*. Setelah itu, maka proses dilanjutkan dengan pembentukan sistem kongruen linier untuk mencari karakter ke-i dari pesan dan mencari solusi dari sistem kongruen linier tersebut dengan menggunakan teorema *Chinese Remainder*. Setelah mendapatkan solusi tersebut, maka dihitung nilai M(i) dengan menggunakan rumus M(i) = M(i)' – rp. Proses diakhiri dengan pengubahan nilai M(i) ke bentuk karakter sehingga diperoleh karakter ke-I dari pesan.

4. Kesimpulan Dan Saran

4.1. Kesimpulan

Dari hasil pengujian yang telah dilakukan pada program dapat ditarik yaitu :
Perancangan Perangkat Lunak Pembelajaran Protokol Secret Sharing Dengan Algoritma Asmuth – Bloom lebih mudah untuk dipelajari dan dimengerti oleh pengguna protocol secret sharing.

4.2. Saran

Beberapa saran pengembangan yang dapat diberikan terhadap perangkat lunak ini adalah Perancangan Perangkat Lunak Pembelajaran Protokol Secret Sharing Dengan Algoritma Asmuth – Bloom perlu di tingkatkan atau perbaikan yang lebih lanjut demi tercapainya proses yang lebih sempurna.

Daftar Pustaka

[1] Ariyus, Dony, 2005, **KRIPTOGRAFI Keamanan Data dan Komunikasi**, Penerbit Graha Ilmu
[2] Munir, R., 2005, **Matematika Diskrit**, Edisi ketiga, Penerbit Informatika Bandung.
[3] Pramono, D., 2002, **Mudah menguasai *Visual Basic 6***, PT. Elex Media Komputindo.
[4] Schneier, B., 1996, ***Applied Crytography***, *Second Edition*, John Willey and Sons Inc..
[5] Suryokusumo, A., 2001, ***Microsoft Visual Basic 6.0***, PT. Elex Media Komputindo.