UNIVERSITAS INDONESIA

PEMBENTUKKAN SKEMA *SECRET SHARING* BERDASARKAN FUNGSI *HASH*

SKRIPSI

SEPTYADI PRABOWO

0806452305

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

PROGRAM STUDI SARJANA MATEMATIKA

DEPOK

JUNI 2012

UNIVERSITAS INDONESIA

PEMBENTUKKAN SKEMA *SECRET SHARING* BERDASARKAN FUNGSI *HASH*

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana sains

SEPTYADI PRABOWO

0806452305

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

PROGRAM STUDI SARJANA MATEMATIKA

DEPOK

JUNI 2012

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

Nama	: Septyadi Prabowo
NPM	: 0806452305
Tanda Tangan	: 
Tanggal	: Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Septyadi Prabowo

NPM : 0806452305

Program Studi : Sarjana Matematika

Judul Skripsi : Pembentukkan Skema *Secret Sharing* berdasarkan Fungsi *Hash*.

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi S1 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.

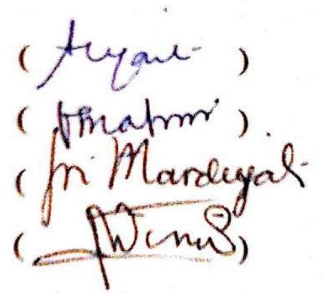
DEWAN PENGUJI

Pembimbing : Dr. Kiki Ariyanti S, M.Si

Penguji I : Bevina D. Handari, Ph.D

Penguji II : Dr. Sri Mardiyanti, M.Kom.

Penguji III : Dra. Yahma Wisnani, M.Kom.



Ditetapkan di : Depok

Tanggal : Juni 2012

KATA PENGANTAR

Alhamdulillahi rabbil aalamiin, segala puji syukur bagi Allah SWT, Tuhan pencipta alam semesta. Atas ridha dan karunia-NYA, penulis telah diberikan kesempatan untuk tetap bertahan menghadapai segala rintangan dan cobaan dalam proses kehidupan yang akhirnya menghantarkan penulis menyelesaikan tugas akhir ini.

Penulis menyadari bahwa masih banyak kekurangan yang terdapat dalam tugas akhir ini. Penulis mengharapkan kritik serta saran guna menyempurnakan tugas akhir ini. Pada kesempatan ini, penulis mengucapakan terima kasih kepada orang-orang yang telah sangat berjasa membantu penulis hingga akhirnya tugas akhir ini dapat terselesaikan dengan baik, terutama kepada :

1. Ibu Dr. Kiki Ariyanti S. selaku Pembimbing tugas akhir yang telah banyak memberi motivasi penulis, menyediakan waktu, memberi saran serta memberikan banyak ilmu yang sangat berharga dan bermanfaat selama penulis menyelesaikan tugas akhir ini.
2. Ibu Saskya Mary Soemartojo, M.Si selaku Pembimbing Akademis penulis yang telah mendukung penulis dalam urusan perkuliahan di setiap semesternya maupun saat penulis mulai menjalankan tugas akhir dan sidang.
3. Bapak Dr. Yudhi Satria selaku Ketua Departemen Matematika, ibu Rahmi Rusin, M. ScTech. selaku Sekretaris Departemen Matematika, ibu Mila Novita, M.Si selaku Koordinator Pendidikan yang banyak membantu penulis dalam urusan perkuliahan maupun organisasi selama penulis menjalani kuliah disini.
4. Seluruh bapak ibu dosen di Departemen Matematika, terutama kepada ibu Suarsih Utama, M.Si., Prof. Dr. Belawati HW, Nora Hariadi, M.Si., Titin Siswatining, DEA., Ida Fithriani, M.Si., Denny Riama Silaban, M.Kom., Dhian Widya, M.Kom., Fevi Novkaniza, M.Si., Helen Burhan, M.Si., Netty Sunandi, M.Si., Dra. Rustina, Siti Aminah, M.Kom., Sri Harini, M.Kom.,

Siti Nurrohmah, M.Si., dan kepada bapak Dr. Alhaji Akbar, Arie Wibowo, M.Si., Prof. Dr. Djati Kerami, Dr. Hengki Tasman, Suryadi Slamet, M.Sc., Suryadi MT, M.T., Dr. Zuherman Rustam, dan semua dosen yang tanpa mengurangi rasa hormat tidak dapat disebutkan satu per satu, terima kasih telah mengajar penulis dari tahun pertama hingga tahun terakhir serta banyak menyumbangkan ilmu pengetahuan baru yang menarik yang belum pernah penulis dapatkan sebelumnya.

5. Seluruh staf tata usaha serta perpustakaan, Mba Santi, Pak Saliman, Pak Ansori, Mas Salman, Mba Rusmi, Pak Turino, Mas Iwan, yang telah banyak membantu seluruh kegiatan penulis selama disini serta Mas Tatang dan Mas Wawan yang banyak membantu saat penulis membutuhkan bantuan.
6. Ibu penulis, Hj. Nurhayati dan Ayah penulis, (Alm.) Marullah yang selalu mendoakan serta mendukung penulis.
7. Kakak penulis, Arif Ali Budiman, dan adik penulis, Siti Farida Wulandari yang selalu memberi semangat kepada penulis.
8. Teman-teman Math'08 yang selalu mendukung penulis, May TA, Danis, Yulial, Purwo, Arkies, Agy, Puput, Dede, Masykur, Juni, Siwi, Vika, Nora, Agnes, Ifah, dan Uchi D., terima kasih selama ini telah mengajak penulis berbagi senyum tawa, pengalaman, dan cerita.
9. Teman-teman seperjuangan tugas akhir, Andi, Adhi, Awe, Arief, Umbu, Maimun, Kiki, Dian, Numa, Risya, Dhila, Tuti, Hindun, Eka, Ega, Risya, Ade, Ines, Luthfa, Sita, Nita, Citra, Cindy, Wulan, Mei, Hendry, Uchi L., Anisah. Terima kasih atas info-info untuk mendukung kelancaran tugas akhir ini.
10. Seluruh kakak angkatan yang bersedia meluangkan waktunya, Kak Ajat, Kak Yanu, Kak Amri, Kak Angga, Kak Michael, Kak Lois, Kak Ar Rizkiyatul, Kak Tino, terima kasih telah menjadi asdos serta aslab yang memberikan ilmu lebih disamping ibu dan bapak dosen disini.
11. Kakak-kakak angkatan 2007, Arif, Hanif, Adit, Anggun, Bowo, Manda, Dita, Ferdy, Nora, Stefi, Anis, dan lainnya. Terima kasih sudah menjadi kakak angkatan yang baik dan banyak memberi bantuan selama ini.

12. Adik-adik angkatan 2009, 2010, dan 2011. Terima kasih sudah menceriakan suasana terutama suasana Hall Math, tetap semangat dan terus berjuang.
13. Teman-teman baik penulis di lingkungan tempat tinggal, Dimas Aditya, Febryan Pasadena, Lita Puspita Sari, terima kasih telah meluangkan waktunya untuk berkumpul bersama.
14. Semua pihak yang telah membantu namun tidak dapat disebutkan satu persatu karena keterbatasan tempat dan daya ingat.

Dan seluruh teman, keluarga yang penulis kenal, baik saat senang maupun susah. Semoga tugas akhir ini menjadi sesuatu yang bermanfaat bagi siapapun. Maaf atas segala kekurangan yang ada. Terima kasih.

Penulis

2012

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini :

Nama	: Septyadi Prabowo
NPM	: 0806452305
Program Studi	: Sarjana Matematika
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	: Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul :

'Pembentukkan Skema *Secret Sharing* berdasarkan Fungsi *Hash*'.

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/ formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 13 juni 2012

Yang menyatakan

(Septyadi Prabowo)

ABSTRAK

Nama : Septyadi Prabowo

Program Studi : Matematika

Judul : Pembentukkan Skema *Secret Sharing* berdasarkan Fungsi *Hash*.

Tugas akhir ini membahas Pembentukkan Skema *Secret Sharing* berdasarkan Fungsi *Hash*. Metode ini menggambarkan bagaimana proses pengamanan suatu pesan/ informasi rahasia dengan membaginya kepada beberapa peserta berupa *share*. Aturan untuk memperoleh pesan rahasia adalah untuk sembarang himpunan bagian yang terdiri dari paling sedikit ***k*** peserta tertentu dapat merekonstruksi kembali pesan, sebaliknya jika kurang dari ***k*** peserta tertentu maka pesan tidak dapat direkonstruksi kembali. Dengan menggunakan fungsi *hash* (*hash function*) dalam proses perhitungan pada skema ini yang mana sulit secara matematis untuk menemukan *preimagenya*, serta penggunaan operasi *XOR* membuat skema ini aman dan effisien dalam hal waktu proses perhitungan. Selain itu, dalam tugas akhir ini juga diberikan ilustrasi bagaimana subbagian pesan/ informasi didistribusikan ke masing-masing peserta, dan proses penemuan kembali pesan/ informasi dalam Skema *Secret Sharing* berdasarkan Fungsi *Hash*.

Kata kunci : skema *secret sharing*, fungsi *hash* (*hash function*), *share*, operasi *XOR*

xiii+57 halaman ; 4 gambar; 3 tabel

Daftar Pustaka : 10 (1979-2011)

ABSTRACT

Name : Septyadi Prabowo

Study Programe : Mathematics

Title : *Secret Sharing Scheme Construction based on* a *Hash Function*.

This final task discusses on this *skripsi* is on the construction of secret sharing schemes based on hash function. This method illustrates how the process of securing secret message/ information by distribute share to several participant. The rule to reconstruct the secret message is as follow: any subset which consists of at least ***k*** certain participant can reconstruct the message, otherwise if less than ***k*** certain participant then the message cannot be reconstructed. By using hash function in the calculation process for this scheme which mathematically difficult to find preimage, and by using *XOR* operation made this scheme is safe and efficient in terms of processing time calculation. Moreover, in this *skripsi* also provided an illustration on how to subsections of message/ information is distributed to each participant, and recovery process of the message/ information of the secret sharing schemes based on hash function.

Key words : *secret sharing scheme, hash function, share, XOR operation*

xiii+57 pages ; 4 pictures; 3 tables

Bibliography : 10 (1979-2011)

DAFTAR ISI

HALAMAN JUDUL. ii

HALAMAN PERNYATAAN ORISINALITAS. iii

HALAMAN PENGESAHAN. iv

KATA PENGANTAR. v

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI. viii

ABSTRAK. ix

ABSTRACT. x

DAFTAR ISI. xi

DAFTAR GAMBAR. xiii

DAFTAR TABEL. xiii

DAFTAR LAMPIRAN. xiii

1. PENDAHULUAN. 1

1.1 Latar Belakang. 1
1.2 Perumusan Masalah dan Ruang Lingkup. 3
1.3 Jenis Penelitian 3
1.4 Tujuan. 3

2. LANDASAN TEORI. 4

2.1 Operasi *XOR*. 4

2.2 Skema *Secret Sharing*. 6

2.2.1 Skema *Secret Sharing* dengan menggunakan Fungsi Polinomial. 7

2.2.2 Skema *Secret Sharing* dengan menggunakan Operasi *XOR*. . . .9

2.4 Aritmatika Modulo. 10

2.3 Fungsi *Hash* 11

3. SKEMA *SECRET SHARING* BERDASARKAN FUNGSI *HASH*. 14

3.1 Algoritma Skema *Secret Sharing* berdasarkan Fungsi *Hash*. 14

3.1.1 Tahap Dealer. 15
3.1.2 Tahap Pembangkit *Pseudo Share*. 17
3.1.3 Tahap Penemuan kembali *Secret*. 19
3.2 Keamanan Skema *Secret Sharing* berdasarkan Fungsi *Hash*. 21
3.2.1 Keamanan *Pseudo Share*. 21
3.2.2 Keamanan *Share*. 22
3.2.3 Keamanan *Secret*. 22
3.3. Sifat-sifat pada Skema *Secret Sharing* berdasarkan Fungsi *Hash*. 23
4. IMPLEMENTASI. 25
5. KESIMPULAN DAN SARAN. 42
4.1 Kesimpulan. 42
4.2 Saran. 44
DAFTAR PUSTAKA. 45
LAMPIRAN. 46

DAFTAR GAMBAR

Gambar 3.1 Ilustrasi untuk tahap Dealer. 16

Gambar 3.2 Ilustrasi untuk tahap pembangkit *Pseudo Share*. 18

Gambar 3.3 Ilustrasi untuk tahap penemuan kembali *Secret*. 20

Gambar 3.4 Ilustrasi proses perhitungan tahap penemuan kembali *Secret* untuk subhimpunan $\boldsymbol{A_{13}}$. 42

DAFTAR TABEL

Tabel 2.1 Tabel Perhitungan Operasi *XOR*. 5

Tabel 4.1 Tabel Hasil Perhitungan tahap Dealer. 28

Tabel 4.2 Tabel Hasil Perhitungan tahap Pembangkit *Pseudo Share*.. 38

DAFTAR LAMPIRAN

Lampiran 1 Tabel ASCII. 46

Lampiran 2 *Source Code* Skema *Secret Sharing* berdasarkan Fungsi *Hash* 48

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kemudahan dalam menjalankan hidup saat ini tidak bisa lepas dari kemajuan dan perkembangan teknologi yang cukup signifikan, misalkan saja dalam hal komunikasi jarak jauh dimana dapat dilakukan dengan mudah dan cepat. Namun disisi lainnya, hal ini juga memiliki beberapa kekurangan dalam aspek keamanan, disamping karena mudahnya orang masa kini untuk mengakses komunikasi jarak jauh seperti *internet*, juga disebabkan masih tingginya pihak-pihak yang tidak bertanggung jawab untuk berusaha memperoleh informasi yang bukan menjadi haknya, dengan maksud dan tujuan tertentu. Salah satu cara untuk mengatasi masalah ini adalah dengan menggunakan kriptografi.

Secara umum, kriptografi adalah ilmu atau seni untuk menjaga kerahasiaan informasi atau pesan (Stalling, W., 2005). Dalam kriptografi terdapat 2 konsep utama, yaitu : *enkripsi* dan *dekripsi*. *Enkripsi* adalah proses dimana data/ informasi yang hendak dikirim diubah menjadi bentuk yang tidak dapat dikenali sebagai informasi awalnya dengan menggunakan suatu algoritma. Sebaliknya, *dekripsi* adalah proses mengubah kembali bentuk tersamar tersebut menjadi informasi awal. Dengan penggunaan kriptografi, kemungkinan pihak yang tidak bertanggung jawab untuk mengetahui informasi yang hendak dikirim dapat diminimalisir.

Salah satu metode dalam kriptografi adalah skema *secret sharing*, yakni teknik membagi informasi yang hendak dikirim menjadi beberapa bagian dan kemudian dengan menggabung bagian-bagian dari informasi tersebut, akan dapat membangun kembali informasi awal (Shamir, A., 1979). Skema ini diperkenalkan oleh Blakely dan Shamir secara terpisah pada tahun 1979. Terdapat 2 aspek penting yang melandasi terbentuknya skema *secret sharing*, yaitu keamanan dan kerahasiaan (Shamir, A., 1979). Keamanan, jika hanya satu orang menyimpan semua *secret*, maka terdapat resiko dimana mungkin orang tersebut menyalahgunakan atau memanipulasi *secret* yang dia pegang seorang diri dengan maksud dan tujuan tertentu. Sementara itu untuk kerahasiaan, mengandung makna jika lebih banyak

orang yang dapat mengakses *secret*, maka kemungkinan *secret* akan bocor menjadi lebih besar.

Skema *secret sharing* memiliki beberapa sifat, diantaranya *perfect*, *verifiable*, dan *ideal*. Skema *secret sharing* bersifat *perfect*, jika *secret* bisa didistribusikan pada himpunan peserta sedemikian sehingga hanya peserta dalam himpunan bagian tertentu saja yang bisa membangun kembali *secret*, sementara peserta dalam himpunan bagian lainnya tidak dapat membangun kembali *secret* (Stinson, D. R., 2002). Skema *secret sharing* dikatakan memiliki sifat *ideal*, jika *share* dan *secret* memiliki ukuran panjang yang sama (Ernest, et al., 1991). Sementara itu, skema *secret sharing* dikatakan bersifat *verifiable*, jika peserta dapat memeriksa kebenaran dari *share* yang mereka terima dari dealer (orang yang membagi *share*) dan membangun kembali *secret* (Stinson, D. R., 2002).

Dalam proses perhitungannya, skema *secret sharing* memiliki beberapa metode. Ide awal metode-metode tersebut adalah sama yaitu membagi *secret* menjadi beberapa bagian, namun untuk metode berbeda memiliki proses perhitungan yang berbeda juga. Beberapa metode perhitungan yang terdapat dalam skema *secret sharing* diantaranya adalah fungsi polinomial, operasi *XOR*, serta fungsi *hash* (yang akan dibahas dalam tugas akhir ini). Secara umum, fungsi *hash* adalah suatu fungsi yang menerima input berupa pesan $\boldsymbol{M}$ dengan panjang sembarang dan menghasilkan suatu output berupa string $\boldsymbol{h}$ dengan panjang tetap (Preneel, B., 1994). Output ini disebut sebagai *hash value* atau *message digest*.

Deskripsi singkat tentang skema ini yaitu untuk mengamankan suatu pesan/ informasi yang hendak dikirim, pertama-tama dealer mengubah pesan/ informasi tersebut menjadi *message digest* dengan panjang yang sudah ditentukan dengan menggunakan suatu fungsi *hash*. Kemudian nilai *message digest* tersebut bersama nilai *pseudo share* yang sebelumnya telah dihitung oleh dealer berdasarkan *shar*e yang telah ditentukan oleh dealer, dikenakan suatu operasi *XOR* sedemikian sehingga menghasilkan *public share* untuk dipublikasikan ke area umum.

Jika sembarang subhimpunan yang terdiri dari beberapa peserta ingin membangun kembali *secret*, maka mereka harus menyerahkan masing-masing *share*

yang mereka miliki ke dealer. Kemudian dealer akan menghitung *pseudo share* berdasarkan *share* yang diterimanya. Nilai *public share* yang mana sebelumnya telah dipublikasi ke area umum oleh dealer akan dikenakan operasi *XOR* bersama dengan nilai *pseudo share*, sehingga menghasilkan *secret* yang merupakan *image* dari pesan yang ingin diperoleh. Dengan perhitungan yang cepat dari fungsi *hash*, serta penggunaan yang sederhana dan efisien seperti memanfaatkan keberadaan fungsi *hash*, menghasilkan suatu skema *secret sharing* yang memiliki sifat *perfect*, *ideal*, *verifiable*, efisien, dan aman (*secure*) (Das, A. & Adhikari, A., 2011).

1.2 Perumusan Masalah dan Ruang Lingkup Masalah

Perumusan masalah yang akan dibahas pada tugas akhir ini adalah bagaimana proses pembentukkan skema *secret sharing* dengan menggunakan fungsi *hash*, proses pembagian *share* dan konstruksi kembali informasi awal.

Sementara, untuk ruang lingkup yang akan dibahas pada tugas akhir ini adalah pembentukkan skema *secret sharing* dengan menggunakan fungsi *hash*.

1.3 Jenis Penelitian

Metode yang digunakan dalam pembuatan skripsi ini adalah dengan menggunakan penelusuran literatur terlebih dahulu, yaitu mencari dan mempelajari beberapa buku, paper, maupun situs *internet* mengenai topik skema *secret sharing* menggunakan fungsi *hash*, kemudian mengimplementasikan skema tersebut kedalam suatu program komputer sederhana.

1.4 Tujuan

Penyusunan skripsi ini bertujuan untuk mempelajari konsep matematis yang melandasi pembentukan algoritma fungsi *hash* dalam skema *secret sharing*, serta perancangan dan implementasinya dalam suatu program komputer sederhana.

BAB 2

LANDASAN TEORI

Pada Bab ini akan diberikan dasar – dasar yang digunakan dalam penulisan tugas akhir ini, yaitu : Operasi *XOR*, Skema *Secret Sharing*, Aritmatika Modulo dan Fungsi *Hash*.

2.1 Operasi *XOR*

Operasi biner *XOR* (*exclusive OR*) merupakan operasi yang digunakan dalam desain sirkuit digital. Simbol dari operasi biner *XOR* (*exclusive OR*) dinyatakan dalam bentuk '$\oplus$' (Simpson, 1987). Pembacaan simbol dari '$\oplus$' adalah '*aut*', dimana '*aut*' berasal dari bahasa latin yang artinya 'atau, tapi bukan keduanya'.

Notasi :

$\oplus$ = operasi *XOR*

$\wedge$ = operasi *AND*

$\vee$ = operasi *OR*

$\sim$ = negasi/ ingkaran

Berdasarkan Simpson (1987) diperoleh hubungan antara operasi *XOR* dengan operasi *AND,* dan operasi *OR* :

$$A \oplus B = (A \sim \wedge B) \wedge \sim (A \wedge B) \qquad \textbf{(2.1)}$$

$$= (A \vee B) \wedge (\sim A \vee \sim B)$$

Operasi biner *XOR* (*exclusive OR*) menerima 2 input untuk setiap proses perhitungannya dan akan menghasilkan sebuah output dari proses tersebut. Operasi *XOR* adalah suatu operasi yang menerima 2 buah input berupa bilangan biner '0' atau '1' dan menghasilkan sebuah output untuk setiap kali proses perhitungannya berupa bilangan biner '0' atau '1'.

Operasi biner *XOR* atau juga dikenal sebagai fungsi *XOR* akan selalu menghasilkan output bernilai ‘1’ untuk setiap proses perhitungannya, jika salah satu dari input yang diberikan bernilai ‘1’. Sebaliknya, operasi biner *XOR* akan menghasilkan output bernilai ‘0’, jika kedua buah input yang diberikan bernilai sama, yaitu ‘0’ atau ‘1’ (Simpson, 1987). Berikut ini tabel output hasil operasi *XOR* untuk setiap 2 buah input yang diberikan.

Tabel 2.1 Tabel perhitungan operasi *XOR*

A	**B**	**A ⊕ B**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

Berdasarkan Simpson (1987), beberapa sifat dari operasi *XOR* diantaranya adalah

1. Komutatif

   $A \oplus B = B \oplus A$

2. Assosiatif

   $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

3. Identitas

   $X \oplus 0 = X$

   $X \oplus \sim X = 1$

   $X \oplus 1 = \sim X$

   $X \oplus X = 0$

2.2 Skema *Secret Sharing*

Kriptografi adalah ilmu atau seni untuk menjaga kerahasiaan informasi atau pesan (Stalling, W., 2005). Dengan penggunaan kriptografi, kemungkinan pihak yang tidak bertanggung jawab untuk memperoleh informasi yang hendak dikirim dapat diminimalisir. Salah satu metode dalam kriptografi adalah skema *secret sharing*, yang diperkenalkan secara terpisah oleh Shamir dan Blakely pada tahun 1979. Terdapat 2 aspek yang melandasi terbentuknya skema ini, yaitu : keamanan dan kerahasiaan (Shamir, A., 1979). Suatu kondisi dimana jika hanya satu orang yang menyimpan *secret*, maka terdapat resiko dimana orang tersebut mungkin menyalahgunakan atau memanipulasi *secret* yang dia pegang sendiri dengan maksud dan tujuan tertentu. Sebaliknya jika lebih banyak orang yang dapat mengakses *secret*, maka kemungkinan *secret* akan bocor menjadi lebih besar.

Skema *secret sharing* (***k***, ***n***) atau yang lebih dikenal sebagai skema ambang (*threshold*) adalah metode pembagian pesan $\boldsymbol{M}$ kepada ***n*** peserta sedemikian sehingga sembarang himpunan bagian yang terdiri dari ***k*** peserta dapat merekonstruksi kembali pesan ***M***, tetapi jika kurang dari ***k*** maka ***M*** tidak dapat direkonstruksi (Shamir, A., 1979). Skema ambang (*threshold*) ini diciptakan atas dasar masalah bagaimana melindungi pesan/ informasi yang hendak dikirim menjadi aman dari pihak yang tidak bertanggung jawab dengan maksud dan tujuan tertentu.

Skema *secret sharing* memiliki beberapa sifat, yaitu ; *perfect*, *verifiable*, dan *ideal*. Skema *secret sharing* dikatakan *perfect*, jika *secret* bisa didistribusikan dalam himpunan peserta sedemikian sehingga hanya himpunan bagian tertentu saja dari himpunan peserta tersebut yang bisa membentuk kembali *secret*, sementara himpunan bagian lainnya tidak dapat membentuk kembali *secret* (Stinson, D. R., 2002). Jadi misalkan terdapat suatu pesan/ informasi $\boldsymbol{M}$, kemudian ***M*** dibagi kedalam ***n*** bagian katakanlah $\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_n$ sedemikian rupa sehingga setiap peserta dari ***n*** peserta tersebut memiliki masing-masing satu buah *share*. Suatu pesan/ informasi ***M*** dapat ditemukan kembali, apabila ***k*** peserta dari ***n*** peserta bergabung bersama dengan masing-masing *share* yang mereka miliki, tetapi untuk kasus dimana hanya kurang dari ***k*** peserta yang bergabung, maka tidak akan diperoleh pesan/ informasi ***M*** tersebut.

Skema *secret sharing* dikatakan *verifiable*, jika peserta dapat memeriksa kebenaran dari *share* yang mereka peroleh dari dealer (Stinson, D. R., 2002). Hal ini bertujuan untuk menghindari manipulasi informasi yang diberikan dealer kepada peserta, dan untuk meminimalisir kemungkinan-kemungkinan terburuk yang dapat mengganggu keamanan dari pesan itu sendiri. Sementara itu, skema *secret sharing* memiliki sifat *ideal* jika *secret* dan *share* memiliki ukuran panjang yang sama (Ernest, et al., 1991).

Dalam pembentukkannya, skema s*ecret sharing* memiliki beberapa metode. Dengan ide awal yang sama yaitu membagi pesan $\boldsymbol{M}$ kedalam beberapa bagian, namun untuk metode berbeda memiliki proses perhitungan yang berbeda. Berikut ini diberikan beberapa proses perhitungan yang ada dalam skema *secret sharing*, yaitu :

1. Fungsi polinomial
2. Operasi *XOR*

2.2.1 Skema *secret sharing* dengan menggunakan fungsi polinomial

Diilustrasikan bahwa terdapat $\boldsymbol{n}$ peserta dan sebuah pesan $\boldsymbol{M}$. Setiap peserta dari ke-$\boldsymbol{n}$ peserta tersebut, masing-masing memiliki kemampuan untuk mengetahui pesan $\boldsymbol{M}$ tersebut, dimana diasumsikan sangat bernilai penting. Masalah muncul, ketika antara peserta satu dengan peserta lainnya tidak saling percaya, dan diinginkan suatu solusi dimana untuk orang tunggal tidak akan bisa mengetahui isi dari pesan $\boldsymbol{M}$ tersebut, atau dengan kata lain dibutuhkan $\boldsymbol{k}$ ($\boldsymbol{k} > 1$) peserta untuk dapat mengetahuinya.

Input pada skema *secret sharing* dengan menggunakan fungsi polinomial adalah pesan $\boldsymbol{M}$. Skema ini didasarkan atas interpolasi polinomial, yaitu untuk membentuk persamaan linear $\boldsymbol{y} = \boldsymbol{s}_0 + \boldsymbol{s}_1\boldsymbol{x}$ dibutuhkan minimal 2 buah titik, $(\boldsymbol{x}_1, \boldsymbol{y}_1)$, dan $(\boldsymbol{x}_2, \boldsymbol{y}_2)$. Sementara untuk membentuk persamaan kuadratik $\boldsymbol{y} = \boldsymbol{s}_0 + \boldsymbol{s}_1\boldsymbol{x} + \boldsymbol{s}_2\boldsymbol{x}^2$ dibutuhkan minimal 3 buah titik, $(\boldsymbol{x}_1, \boldsymbol{y}_1)$, $(\boldsymbol{x}_2, \boldsymbol{y}_2)$, dan $(\boldsymbol{x}_3, \boldsymbol{y}_3)$. Sehingga jika diberikan $\boldsymbol{k}$ titik dalam bidang 2 dimensi $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots\ldots, (\boldsymbol{x}_k, \boldsymbol{y}_k)$ dimana $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ ($\boldsymbol{i} \neq \boldsymbol{j}$), maka dari $\boldsymbol{k}$ titik tersebut dapat diperoleh sebuah polinomial $\mathbf{s(x)} = \boldsymbol{s}_0 + \boldsymbol{s}_1\boldsymbol{x} + \boldsymbol{s}_2\boldsymbol{x}^2 + \ldots + \boldsymbol{s}_{\boldsymbol{k-1}}\boldsymbol{x}^{\boldsymbol{k-1}}$ dengan derajat $\boldsymbol{k}$-1 sedemikian sehingga $\boldsymbol{s}(\boldsymbol{x}_i) = \boldsymbol{y}_i$ untuk $\boldsymbol{i} = 1, \ldots \boldsymbol{k}$.

Berdasarkan Adi Shamir (1979), berikut ini dijelaskan algoritma skema *secret sharing* dengan menggunakan fungsi polinomial.

a. Pilih bilangan prima $\boldsymbol{p}$, dimana harus lebih besar dari jumlah pesan $\boldsymbol{M}$ dan juga lebih besar dari jumlah $\boldsymbol{n}$ peserta. Semua perhitungan dihasilkan dalam modulo $\boldsymbol{p}$.

b. Pilih secara acak $\boldsymbol{k}$ – 1 buah bilangan bulat dalam modulo $\boldsymbol{p}$, misalkan $\boldsymbol{s_1}$, $\boldsymbol{s_2}$, …, $\boldsymbol{s_{k-1}}$, dan nyatakan polinomial :

$$s(\mathbf{x}) \equiv \boldsymbol{s}_0 + \boldsymbol{s_1 x} + \boldsymbol{s_2} x^2 + \ldots + \boldsymbol{s_{k-1}}\, \boldsymbol{x}^{k-1} \pmod{\boldsymbol{p}} \qquad \textbf{(2.2)}$$

sedemikian sehingga $\boldsymbol{s}\,(\mathbf{0}) \equiv \boldsymbol{s}_0 \pmod{\boldsymbol{p}}$

c. Untuk $\boldsymbol{n}$ peserta, pilih bilangan bulat berbeda, $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n} \pmod{\boldsymbol{p}}$

dan setiap orang memperoleh *share* ($\boldsymbol{x_i}, \boldsymbol{y_i}$) yang dalam hal ini $\boldsymbol{y}_i \equiv \boldsymbol{s}_i(\boldsymbol{x}_i) \pmod{\boldsymbol{p}}$. Misalkan, untuk $\boldsymbol{n}$ orang dipilih $\boldsymbol{x_1} = 1$, $\boldsymbol{x_2} = 2$, …, $\boldsymbol{x_n} = \boldsymbol{n}$.

d. Misalkan $\boldsymbol{k}$ peserta akan merekonstruksi pesan $\boldsymbol{M}$ dengan *share* masing-masing yang mereka miliki yaitu ($\boldsymbol{x_1}, \boldsymbol{y_1}$), ($\boldsymbol{x_2}, \boldsymbol{y_2}$) …, ($\boldsymbol{x_k}, \boldsymbol{y_k}$).

mensubstitusikan setiap ($\boldsymbol{x_t}, \boldsymbol{y_t}$) ke dalam polinomial:

$$\boldsymbol{S}\,(\boldsymbol{x}) \equiv \boldsymbol{s}_0 + \boldsymbol{s_1}\, \boldsymbol{x} + \boldsymbol{s_2} x^2 + \ldots + \boldsymbol{s_{k-1}}\, \boldsymbol{x}^{k-1} \pmod{\boldsymbol{p}}$$

Ini berarti: $\boldsymbol{y_t} \equiv \boldsymbol{s}_0 + \boldsymbol{s_1}\, \boldsymbol{x_t^1} + \ldots + \boldsymbol{s_{t-1}} \boldsymbol{x_t^{k-1}} \pmod{\boldsymbol{p}}$, $1 \leq \boldsymbol{t} \leq \boldsymbol{k}$

e. Jika dimisalkan $\boldsymbol{s_0} = \boldsymbol{M}$, maka dapat ditulis ulang sistem persamaan ke dalam bentuk matriks :

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^{t-1} \\ 1 & x_2 & \ldots & x_2^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_t & \ldots & x_t^{t-1} \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{t-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{pmatrix}$$

Dengan eliminasi Gaus Jordan, diperoleh $\boldsymbol{s_0} = \boldsymbol{M}$, dimana merupakan *secret* yang ingin diperoleh (Shamir, A., 1979).

2.2.2 Skema *secret sharing* dengan menggunakan operasi *XOR*

Dengan ilustrasi yang sama, yaitu bahwa terdapat n peserta dan sebuah pesan M. Setiap peserta dari ke-n peserta tersebut, masing-masing memiliki kemampuan untuk mengetahui pesan M dimana diasumsikan sangat bernilai penting. Masalah muncul, ketika antara peserta satu dengan peserta lainnya tidak saling percaya. Salah satu solusi untuk menjaga keamanan dari pesan M berdasarkan masalah tersebut adalah untuk dapat mengetahui isi pesan M dibutuhkan k ($k > 1$) peserta, sehingga untuk orang tunggal tidak akan bisa mengetahui pesan M tersebut.

Input pada skema *secret sharing* dengan menggunakan operasi *XOR* adalah pesan M. Skema ini didasarkan atas dasar operasi *XOR*, yaitu suatu operasi yang menerima 2 buah input berupa bilangan biner dan menghasilkan sebuah output berupa bilangan biner untuk setiap prosesnya.

Berdasarkan Kurihara, et al., (2008), berikut ini dijelaskan Algoritma distribusi dari skema *secret sharing* dengan menggunakan operasi *XOR*.

a. Pilih bilangan prima n_p, dimana n_p harus lebih besar dari jumlah pesan M dan juga lebih besar dari jumlah n peserta.
b. Bagi pesan M kedalam (n_p-1) bagian, dimana setiap bagian memiliki d-bit dengan ukuran yang sama.

   $$M_i , i = 1, 2, \ldots ,(n_p\text{-}1).$$

   didefinisikan M_0 dimana memiliki panjang dengan ukuran d-bit berupa barisan 0.

c. Bangkitkan { ($k - 1$) n_p} - 1 d-bit bilangan acak secara bebas dari masing-masing lainnya dengan probabilitas seragam, $r^h_{h \cdot i+j}$

   dimana $h = 0, \ldots,(k - 2),$ $i = 0,1, \ldots, (n_p - 1),$

   $j = 0,1 \ldots, (n_p - 2).$

   setiap $r^h_{(h \cdot i)+j}$ memiliki ukuran d-bit yang sama.

d. Jalankan operasi *XOR* dengan persamaan berikut

$$\boldsymbol{W_{(i,j)} = \left(\bigoplus_{h=0}^{k-2} r_{(h\cdot i)+j}^{h}\right) \oplus s_{j-i}} \qquad \textbf{(2.3)}$$

dimana, $\boldsymbol{i}$ = 0,1, . . , ($\boldsymbol{n_p}$ – 1),

$\boldsymbol{j}$ = 0,1 . . ., ($\boldsymbol{n_p}$ – 2).

e. Menggabungkan ($\boldsymbol{n_p}$ – 1) bagian dan membangkitkan sebuah share $\boldsymbol{w_i}$ untuk diberikan ke setiap peserta $\boldsymbol{P_i}$

$$\boldsymbol{w_i = w_{(i,1)} \| w_{(i,2)} \| \ldots\ldots \| w_{(i,j)}}$$

dimana, $\boldsymbol{i}$ = 0,1, . . , ($\boldsymbol{n_p}$ – 1),

$\boldsymbol{j}$ = 0,1 . . ., ($\boldsymbol{n_p}$ – 2).

2.3 Aritmatika Modulo

Dalam beberapa kasus pembagian untuk sembarang bilangan bulat, terkadang ditemukan suatu kondisi dimana bilangan bulat tersebut mungkin tidak habis dibagi dengan bilangan bulat lainnya, atau dengan kata lain terdapat sisa dalam proses pembagian tersebut. Berikut ini definisi yang berkaitan dengan aritmatika modulo berdasarkan Kenneth H. Rosen (1999), yaitu : Misalkan ***a*** adalah sembarang bilangan bulat dan ***m*** adalah bilangan bulat positif, maka ***a*** mod ***m*** menunjukkan sisa pembagian ketika sembarang bilangan bulat ***a*** dibagi oleh sembarang bilangan bulat positif ***m***. jika bilangan bulat ***r*** menyatakan sisa pembagian dari pembagian tersebut, maka diperoleh kondisi $\boldsymbol{a = q\,m + r}$ dimana $0 \leq \boldsymbol{r} < \boldsymbol{m}$.

Contoh : 17 mod 5 = 2 (karena 17 = 3 (5) + 2)

2.4 Fungsi *Hash*

Fungsi *hash* diperkenalkan dalam kriptografi pada tahun 70-an akhir sebagai sebuah metode untuk melindungi keaslian dari informasi (Preenel, B., 1994). Seiring dengan bertambahnya waktu, kemudian fungsi *hash* berkembang menjadi suatu metode perlindungan yang efektif dalam berbagai masalah dalam telekomunikasi dan jaringan komputer.

Perkembangan saat ini dalam aspek telekomunikasi telepon genggam (*hand phone*) seperti sistem GSM memungkinkan komunikasi dapat berlangsung dimanapun, dan kapanpun secara bebas baik dirumah, kantor, atau dijalanan. Pesan elektronik seperti email telah menjadi suatu cara yang disukai dalam berkomunikasi masa kini, karena kemudahan serta keefektifan waktu yang digunakan dalam proses pengiriman pesan/ informasi. Pada waktu yang sama, EDI (*electronic data interchange*) diperkenalkan sebagai suatu metode untuk menyampaikan proses informasi secara otomatis dalam suatu perusahaan, antara penyedia dan klien kedalam sebuah sistem tunggal. Selain itu, aplikasi lainnya seperti *home banking* menjadi sangat terkenal sebagai suatu metode untuk proses pembelanjaan *online* yang dilakukan secara tidak langsung. Perkembangan telekomunikasi saai ini menuntut terciptanya sistem keamanan baru yang tentunya harus lebih aman.

Fungsi *hash* dirancang dengan tujuan utama yakni melindungi keaslian dari suatu pesan/ informasi. Keberadaan fungsi *hash* yang aman dan efisien dapat digunakan dalam berbagai aplikasi, misalkan saja digunakan sebagai fungsi acak karena untuk input dengan panjang sembarang akan menghasilkan output dengan panjang tetap. Berdasarakan Preenel, B. (1994), beberapa aplikasi penting dari keberadaan fungsi *hash*, yaitu ; perlindungan dari *pass-phrases* (*pass-phrases* merupakan suatu *password* dengan panjang sembarang), konstruksi skema efisien *digital signature*, dan konstruksi dari suatu algoritma *enkripsi*. Fungsi *hash* adalah suatu fungsi yang beroperasi pada sebuah string input dengan panjang tidak tentu atau sembarang, dan menghasilkan string output dengan panjang yang sudah ditentukan (Preenel, B., 1994). Output dari suatu fungsi *hash* disebut sebagai *hash value*, atau *message digest*.

Fungsi *hash* bersifat satu arah, yaitu jika diberikan sebuah ***y*** hasil dari fungsi **H**, maka akan sulit (secara perhitungan) untuk menemukan sebuah ***x*** dimana **H** (***x***) = ***y*** (*preimage resistant*). Selain itu, jika diberikan ***x*** dan **H** (***x***) maka akan tidak mungkin (secara perhitungan) untuk menemukan sebuah $x' \neq x$ dimana **H** (***x'***) = **H** (***x***) (*second preimage*) (Preneel, B., 1994).

Berdasarkan Bart Preneel (1994), fungsi *hash* **H** harus dapat meminimalkan terjadinya tubrukan (*collision resistance*), yaitu sulit untuk menemukan 2 pesan berbeda yang mana memiliki nilai fungsi *hash* yang sama. Selain itu, suatu fungsi *hash* harus memiliki sifat mudah untuk dihitung yang berguna untuk keefisiensian dalam waktu proses perhitungan.

Seperti yang telah dijelaskan sebelumnya bahwa input dalam fungsi *hash* adalah suatu pesan ***M*** dengan panjang sembarang. Namun, karena proses perhitungan berlangsung dalam suatu mesin komputer, maka pesan atau data yang ingin dibuat *message digest* harus dikodekan terlebih dahulu sedemikian rupa sehingga pesan tersebut dapat diterima oleh komputer sebagai suatu input.

Komputer yang dibuat oleh berbagai pabrik membuat suatu kesepakatan untuk melambangkan sebuah huruf atau angka, meski tidak semua pabrik setuju bersatu untuk melambangkannya, sehingga ada yang sepakat untuk menggunakan kode ASCII (*American Standart Code for Information Interchange*), ataupun ada yang setuju dengan menggunakan kode EBCDIC (*Extend Binary Code Decimal Interchange Code*), dan sebagainya. Kode standar Amerika untuk pertukaran informasi atau ASCII (*American Standart Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode*, tetapi ASCII lebih bersifat *universal*. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit, dimulai dari ‘00000000’ hingga ‘11111111’. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 225 dalam sistem bilangan desimal.

Suatu pesan terdiri dari beberapa huruf atau angka, misalkan kata ‘Universitas’ terdiri dari 11 huruf yang masing-masing huruf memiliki kode ASCII. Berdasarkan tabel ASCII (pada lampiran), diperoleh :

- U = 85
- n = 110
- i = 105
- v = 118
- e = 101
- r = 114
- s = 115
- i = 105
- t = 116
- a = 97
- s = 115

Salah satu contoh dari fungsi *hash* adalah fungsi modulo. Misalkan jika pesan dengan kata ‘Universitas’ tersebut menjadi input dari fungsi *hash* modulo, kemudian ditentukan pembagi dari fungsi tersebut yaitu bilangan prima 7. Pesan dengan kata ‘Universitas’ memiliki kode ASCII yang secara keseluruhan dari penjumlahan huruf-hurufnya adalah 1181, selanjutnya nilai penjumlahan tersebut dibagi dengan bilangan prima 7 menghasilkan sisa pembagian dengan nilai 5. Hasil modulo tesebut kemudian diubah menjadi suatu bilangan biner dengan panjang 8, dengan proses perhitungan sebagai berikut ;

5/2 = 2 (1)
2/2 = 1 (0) } 0 0 0 0 0 1 0 1
1/2 = 0 (1)

Sehingga untuk input berupa pesan ‘Universitas’, dengan perhitungan fungsi *hash* modulo dihasilkan output berupa bilangan biner 0 0 0 0 0 1 0 1.

BAB 3

SKEMA *SECRET SHARING* BERDASARKAN FUNGSI *HASH*

Berdasarkan Das, A., & Adhikari, A. (2011), pada bab 3 ini akan dibahas mengenai Algoritma dari skema *secret sharing* berdasarkan fungsi *hash*, beserta keamanan skema, dan sifat-sifat dari skema tersebut.

3.1 Algoritma Skema *Secret Sharing* berdasarkan Fungsi *Hash*

Salah satu metode dalam skema *secret sharing* adalah menggunakan fungsi *hash*, yaitu suatu fungsi yang beroperasi pada sebuah string input dengan panjang sembarang, dan menghasilkan string output dengan panjang yang sudah ditentukan. Output ini biasa dikenal dengan sebutan *hash value* atau *message digest* (Preneel,B., 1994). Fungsi *hash* menerima input berupa pesan $\boldsymbol{M}$ dengan panjang sembarang dan menghasilkan output berupa string $\boldsymbol{h}$ dengan panjang tetap.

Dalam skema ini, fungsi *hash* yang digunakan adalah fungsi *hash* modulo, dimana mengambil input dengan panjang sembarang dan akan menghasilkan output dengan panjang yang sudah ditentukan sebelumnya, dalam hal ini dengan panjang 8 bit . Algoritma ini juga memperkenalkan struktur akses yang menggunakan prinsip ambang (*threshold*) yang bersesuaian, dimana dealer dapat menetapkan siapa saja yang dapat mengakses *secret*.

- **<u>Notasi</u>**

  P : himpunan peserta

  $\boldsymbol{D}$: Dealer, yaitu orang yang bertugas membagi *share* ke peserta.

  $\boldsymbol{n}$: banyaknya peserta

  $\boldsymbol{M_i}$: pesan ke-$\boldsymbol{i}$ dimana $\boldsymbol{i}$ = 1,2, . . . $\boldsymbol{z}$.

  $\boldsymbol{s_i}$: *secret* ke-$\boldsymbol{i}$ yang menunjukkan nilai *hash* dari pesan ke-$\boldsymbol{i}$

  ($\boldsymbol{s_i}$ = H($\boldsymbol{M_i}$))

$\boldsymbol{A_{il}}$: subhimpunan peserta ke-*l* untuk *secret* ke-*i*

$\Gamma\boldsymbol{s_i}$: Struktur akses yaitu himpunan peserta yang dapat merekonstruksi *secret* ke-*i*

$\boldsymbol{x_\alpha}$: *share* untuk peserta ke-$\boldsymbol{\alpha}$, dimana $\boldsymbol{\alpha}$ = 1, 2, . . .$\boldsymbol{n}$

H : fungsi *hash* modulo

⌊ ⌋ : fungsi *floor*

Berdasarkan Das, A., & Adhikari, A. (2011), Algoritma dari skema *secret sharing* berdasarkan fungsi *hash* adalah sebagai berikut.

Diasumsikan : Dealer $\boldsymbol{D}$ adalah orang yang terpercaya.

Misalkan $\boldsymbol{s_1}$, $\boldsymbol{s_2}$, . . ., $\boldsymbol{s_z}$ adalah $\boldsymbol{z}$ *secret* untuk dibagikan oleh dealer terpercaya $\boldsymbol{D}$ sedemikian rupa sehingga $\boldsymbol{s_i} \in \{0,1\}^8$ untuk $\boldsymbol{i}$ = 1,2, . . ., $\boldsymbol{z}$, dengan struktur akses $\Gamma\boldsymbol{s_i}$ = { $\boldsymbol{A_{i1}}$, $\boldsymbol{A_{i2}}$, ..., $\boldsymbol{A_{it_i}}$}, dimana $\{0,1\}^{\boldsymbol{8}}$ menunjukkan himpunan semua string biner dengan panjang tetap 8 dan $\boldsymbol{t_i}$ = | $\Gamma\boldsymbol{s_i}$|, menyatakan panjang dari himpunan peserta yang dapat merekonstruksi *secret* ke-*i*.

Skema terdiri dari 3 tahap, yaitu tahap Dealer, tahap pembangkit *Pseudo Share*, dan tahap *Recovery Secret* (Penemuan Kembali *Secret*).

3.1.1 Tahap Dealer

- Langkah 1 : Pemilihan

  Pilih fungsi *hash* modulo **H** yang aman dan dapat meminimalkan terjadinya tubrukan (*collision*) (Preneel, B., 1994). Fungsi *hash* modulo **H** mengambil input berupa string dengan panjang sembarang, kemudian menghasilkan output berupa bilangan biner dengan panjang yang sudah ditentukan sebelumnya, dalam hal ini dengan panjang 8 bit.

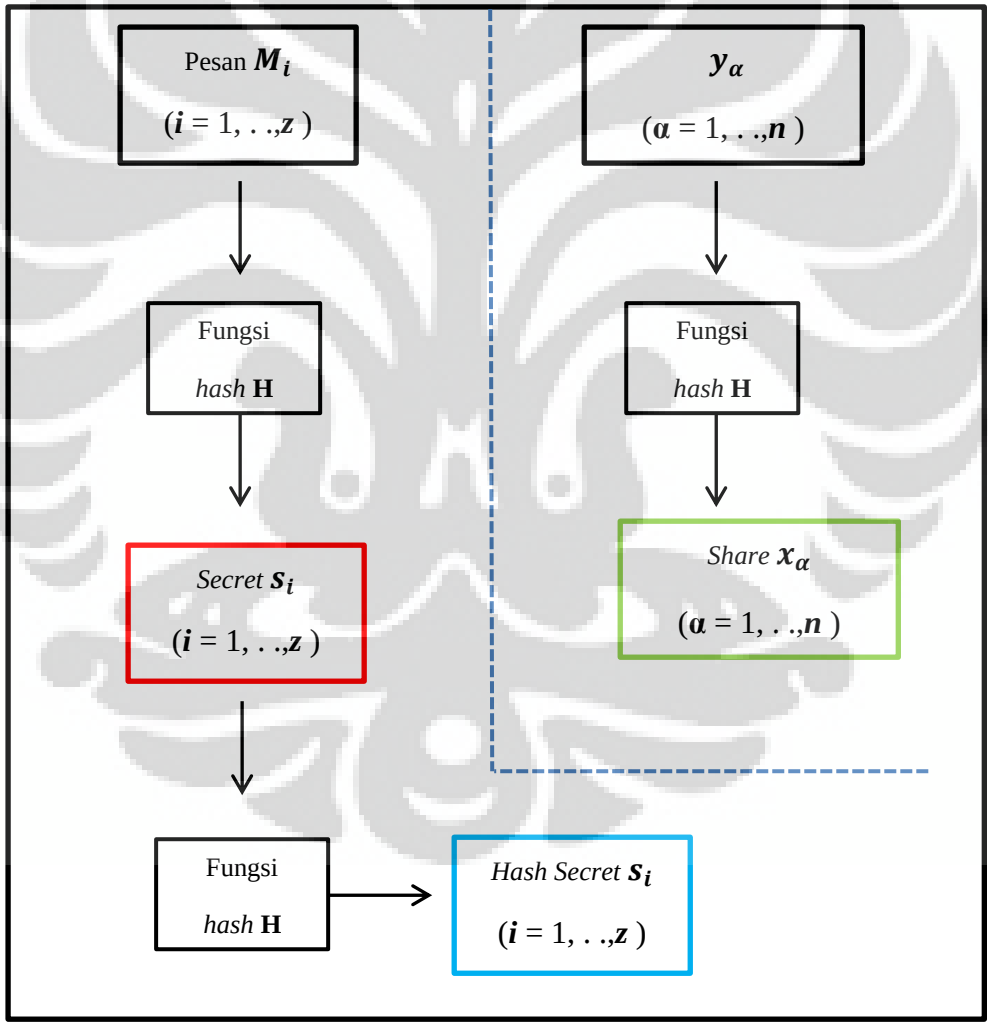
  Misalkan terdapat $\boldsymbol{z}$ pesan rahasia $\boldsymbol{M_i}$, $\boldsymbol{i}$ = 1,2, . . . $\boldsymbol{z}$.

  Dealer $\boldsymbol{D}$ menghitung $\boldsymbol{s_i}$ = **H** ($\boldsymbol{M_i}$), untuk mendapatkan nilai *hash* $\boldsymbol{s_i}$ dari pesan rahasia $\boldsymbol{M_i}$ ($\boldsymbol{i}$ = 1,2, . . . $\boldsymbol{z}$).

- Langkah 2 : pengiriman *share*

Dealer ***D*** mengirim secara rahasia *share* x_α dengan panjang 8 bit ke setiap peserta (x_α = **H** (y_α), untuk α = 1, 2, . .n), kemudian mempublikasikan fungsi *hash* modulo **H**, dan himpunan peserta yang dapat merekonstruksi *secret* ke-*i* Γs_i (untuk i = 1, 2, . . .z) ke area umum. Bilangan $y_\alpha \in_R \{0,1\}$, α = 1, 2, . . .n, dimana '$\in_R$' menunjukkan pemilihan acak. (H (y_α) = x_α).

Dari pembahasan tentang langkah-langkah untuk tahap dealer dari skema *secret sharing* berdasarkan fungsi *hash*, langkah-langkah tersebut dapat diringkas dalam ilustrasi berikut.



Gambar 3.1 Ilustrasi proses perhitungan untuk Tahap Dealer

3.1.2 Tahap pembangkit *Pseudo share*.

Misalkan $\boldsymbol{l} = \lfloor \boldsymbol{log_2 z} \rfloor + 1$ dan $\boldsymbol{m} = \lfloor \boldsymbol{log_2 t} \rfloor + 1$, dimana $\boldsymbol{t}$ = max { $\boldsymbol{t_1}, \boldsymbol{t_2}, . . , \boldsymbol{t_z}$} dan $\boldsymbol{t_i} = |\Gamma \boldsymbol{s_i}|$.

- Langkah 1 :

  Untuk $\boldsymbol{i}$ = 1,2, . . ,$\boldsymbol{z}$; $\boldsymbol{j}$ = 1, 2, . ., $\boldsymbol{t_i}$; dealer menghitung nilai *Public Share* $\boldsymbol{S_{ij}}$

  $$\boldsymbol{S_{ij}} = \boldsymbol{s_i} \oplus \{\oplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\} \quad \textbf{(3.1)}$$

  dimana $\boldsymbol{i_l}$ menunjukkan $\boldsymbol{l}$ bit perwakilan biner ke-$\boldsymbol{i}$, $\boldsymbol{j_m}$ menunjukkan $\boldsymbol{m}$-bit perwakilan biner ke $\boldsymbol{j}$, '$\lfloor \rfloor$' menunjukkan fungsi *floor*, '| |' menunjukkan rangkaian dari 2 string biner dan '$\oplus$' menunjukkan operasi *XOR* antara 2 buah string biner.

  nilai $\boldsymbol{s_i}$ adalah *secret* ke-$\boldsymbol{i}$ yang merupakan nilai *hash* untuk pesan $\boldsymbol{M_i}$. *Secret* $\boldsymbol{s_i}$ merupakan output dengan panjang 8 bit hasil fungsi *hash* **H**, yang mana mengambil input berupa pesan $\boldsymbol{M_i}$ dengan panjang sembarang ($\boldsymbol{s_i}$ = **H** ($\boldsymbol{M_i}$), untuk $\boldsymbol{i}$ = 1, . ., $\boldsymbol{z}$)

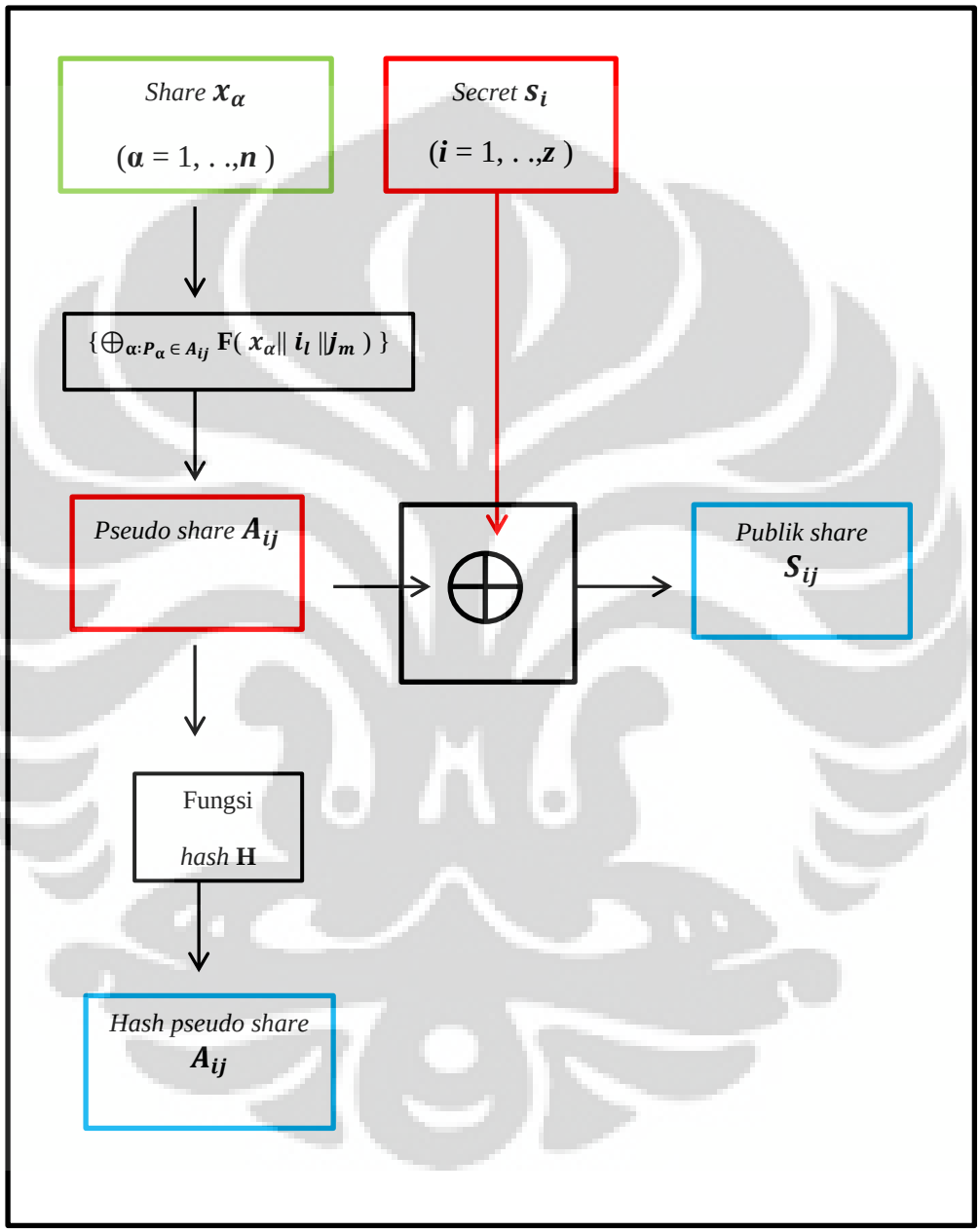
  fungsi **F**($\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m}$) mengambil input dengan panjang sembarang dan akan menghasilkan suatu ouput untuk setiap inputnya berupa bilangan biner dengan panjang 8 bit, yang diambil dengan *high-order*.

- Langkah 2 :

  Delaer $\boldsymbol{D}$ mempublikasikan ke area umum ;

  - Fungsi *hash* modulo **H**.
  - Struktur Akses $\Gamma \boldsymbol{s_i}$, untuk $\boldsymbol{i}$ = 1, 2, . . .$\boldsymbol{z}$.
  - **H** ($\boldsymbol{s_i}$) adalah nilai *hash* dari *secret* ke-$\boldsymbol{i}$, untuk $\boldsymbol{i}$ = 1, . . $\boldsymbol{z}$.
    (**H**($\boldsymbol{s_i}$) = **H** (**H**($\boldsymbol{M_i}$))
  - **H**{$\oplus_{\alpha:P_\alpha \in A_{ij}}$ **F**($\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m}$) } adalah nilai *hash* dari *pseudo share* untuk subhimpunan ke- $\boldsymbol{j}$, *secret* ke- $\boldsymbol{i}$.
  - Nilai *public share* $\boldsymbol{S_{ij}}$, menandakan *public share* dari subhimpunan ke- $\boldsymbol{j}$ untuk *secret* ke- $\boldsymbol{i}$.

Dari pembahasan tentang langkah-langkah untuk tahap pembangkit *pseudo share* dari skema *secret sharing* berdasarkan fungsi *hash*, maka langkah-langkah tersebut dapat diringkas dalam ilustrasi berikut.



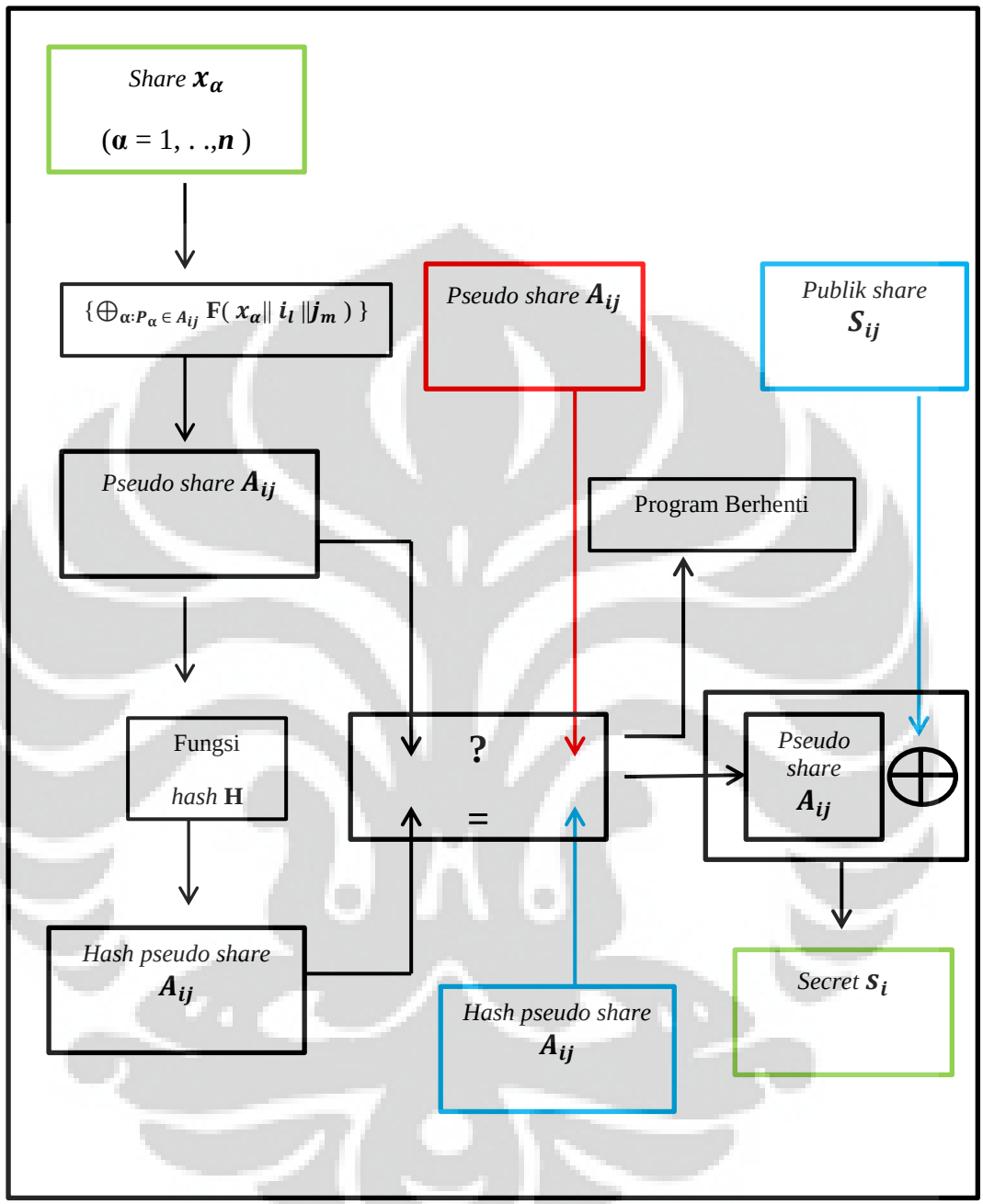
Gambar 3.2 Ilustrasi proses perhitungan untuk Tahap pembangkit *Pseudo Share*

3.1.3 Tahap penemuan kembali *Secret*

Misalkan subhimpunan $\boldsymbol{A_{ij}}$ dari struktur akses $\Gamma \boldsymbol{s_i}$ ingin memperoleh pesan $\boldsymbol{M_i}$. Selanjutnya masing-masing peserta dari subhimpunan tersebut berkumpul dan menyerahkan *share* masing-masing yang mereka miliki ke dealer. Dengan menggunakan fungsi $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$, kemudian dealer menghitung untuk mendapatkan nilai *pseudo share* berdasarkan *share* yang diberikan masing-masing peserta dari subhimpunan tersebut. Setelah nilai *pseudo share* didapatkan, kemudian dealer memverifikasi kebenaran nilai *pseudo share* yang diperoleh berdasarkan input *share* masing-masing peserta dengan nilai *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ yang mana sebelumnya dealer telah menghitung dan menyimpannya, dan juga menggunakan nilai *hash pseudo share* $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ yang sebelumnya telah dipublikasikan ke area umum oleh dealer.

Jika nilai *pseudo share* tersebut adalah benar, maka tahap selanjutnya dealer menghitung $\boldsymbol{s_i} = \boldsymbol{S_{ij}} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$, dimana $\boldsymbol{S_{ij}}$ adalah nilai *public share* yang sebelumnya telah dipublikasikan ke area umum oleh dealer, dan $\boldsymbol{s_i}$ adalah nilai *hash* dari pesan ke-$\boldsymbol{i}$ ($\boldsymbol{s_i} = \mathbf{H}(\boldsymbol{M_i})$). Setelah nilai *hash* dari pesan ke-$\boldsymbol{i}$, $\boldsymbol{s_i}$ diperoleh, kemudian dealer mencocokan nilai $\boldsymbol{s_i}$ tersebut dengan *preimage* dari nilai tersebut yang merupakan pesan ke-$\boldsymbol{i}$ ($\boldsymbol{M_i}$), dimana dealer telah menghitung dan menyimpan sebelumnya. Setelah pesan ke-$\boldsymbol{i}$ ($\boldsymbol{M_i}$) diketahui, kemudian dealer memberikan pesan rahasia ke-$\boldsymbol{i}$ ($\boldsymbol{M_i}$) tersebut ke masing-masing peserta dari subhimpunan tersebut, dan peserta dapat memverifikasi kebenaran dari pesan tersebut dengan nilai $\mathbf{H}(\mathbf{H}(\boldsymbol{M_i})) = \mathbf{H}(\boldsymbol{s_i})$ yang sebelumnya telah dipublikasikan dealer ke area umum.

Dari pembahasan tentang langkah-langkah untuk tahap penemuan kembali *secret* dari skema *secret sharing* berdasarkan fungsi *hash*, langkah-langkah tesebut dapat diringkas dalam ilustrasi berikut.



Gambar 3.3 Ilustrasi proses perhitungan untuk Tahap penemuan kembali *Secret*

3.2 Keamanan Skema *Secret Sharing* berdasarkan Fungsi *Hash*

Skema *secret sharing* pada tugas akhir ini menggunakan fungsi *hash* sederhana, yakni fungsi modulo. Untuk input dengan panjang sembarang, menghasilkan output dengan panjang tetap, dalam hal ini 8 bit. Tidak adanya operasi seperti *eksponensial* dan *inversion* pada skema ini, serta hanya penggunaan fungsi *hash* modulo dan operasi *XOR*, menghasilkan waktu perhitungan yang cepat (Das, A. & Adhikari, A., 2011).

Keamanan dari *pseudo share*, *share*, dan *secret*, menyebabkan skema ini menjadi aman secara keseluruhan, serta kemungkinan terjadinya tubrukan (*collision*) dapat diminimalisir.

3.2.1 Keamanan *Pseudo Share*

Misalkan salah satu peserta $\boldsymbol{P_\alpha}$ untuk $\boldsymbol{\alpha}$ = 1, . . ,$\boldsymbol{n}$ dari subhimpunan $\boldsymbol{A_{ij}}$ berusaha untuk memperoleh *secret* $\boldsymbol{s_i}$ dengan *share* $\boldsymbol{x_\alpha}$ yang dia miliki seorang diri. Kemudian dia menyerahkan *share* yang dia miliki ke dealer, lalu dealer menghitung nilai *pseudo share* berdasarkan *share* yang diperoleh dari peserta tersebut melalui fungsi *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha}\| \boldsymbol{i_l} \|\boldsymbol{j_m})\}$. Masalah muncul ketika nilai *pseudo share* yang diperoleh tidak sesuai dengan nilai *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha}\| \boldsymbol{i_l} \|\boldsymbol{j_m})\}$ yang mana sebelumnya dealer telah menghitungnya dan nilai *hash pseudo share* $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha}\| \boldsymbol{i_l} \|\boldsymbol{j_m})\}$ yang mana sebelumnya telah dipublikasi oleh dealer ke area umum. Hal ini disebabkan karena pada proses penginputan fungsi *pseudo share*, input yang dimasukkan tidak sesuai dengan yang ditetapkan dealer yakni input yang harus dimasukkan sebanyak $\boldsymbol{k}$ ($\boldsymbol{k} > 1$) nilai yang bersesuaian untuk sekali proses penginputan, sementara dia hanya seorang diri maka *share* yang diberikan ke dealer juga hanya satu. Karena antara nilai *pseudo share* berdasarkan input nilai *share* peserta tersebut dengan nilai *pseudo share* yang sebelumnya telah dihitung dan disimpan oleh *dealer* dan juga nilai *hash* dari nilai tersebut tidak sesuai, maka dealer tidak dapat melanjutkan proses perhitungan untuk mendapatkan *secret* $\boldsymbol{s_i}$. Proses penginputan ini mengikuti kaidah Shamir, yaitu sembarang himpunan bagian yang terdiri dari $\boldsymbol{k}$ ($\boldsymbol{k} > 1$) peserta yang ditentukan dapat

merekonstruksi kembali pesan $\boldsymbol{M}$, tetapi jika kurang dari $\boldsymbol{k}$ bersesuaian maka $\boldsymbol{M}$ tidak dapat direkonstruksi. Jadi, untuk mendapatkan *secret* $\boldsymbol{s_i}$, peserta tersebut harus mencari $\boldsymbol{k}$-1 nilai *share* lainnya yang bersesuaian sedemikian rupa *secret* $\boldsymbol{s_i}$ dapat dikonstruksi kembali. Dalam algoritma ini bahkan dapat ditentukan $\boldsymbol{k}$ peserta tertentu yang tidak dapat mengakses *secret*.

Dengan kasus yang hampir sama namun dengan subjek yang berbeda yaitu pada pihak asing. Agar pihak asing tersebut bisa memperoleh *secret* $\boldsymbol{s_i}$, maka dia membutuhkan *share* $\boldsymbol{x_\alpha}$ sebanyak $\boldsymbol{k}$ yang bersesuaian, dimana diasumsikan bahwa pihak asing tesebut tidak mengetahui atau memiliki satupun *share* $\boldsymbol{x_\alpha}$.

3.2.2 Keamanan *Share*

Setiap kata atau kalimat terdiri dari beberapa karakter seperti huruf, angka, ataupun simbol-simbol, yang mana setiap karakter tersebut memiliki bilangan ASCII. Perhitungan *share* $\boldsymbol{x_\alpha}$ menggunakan fungsi *hash* modulo, yang mana hasil dari penjumlahan bilangan ASCII untuk setiap karakter dari kata atau kalimat yang diinput, kemudian dikenakan fungsi modulo. Hasil dari fungsi modulo tersebut dikonversikan kedalam bilangan biner dengan panjang 8. Setiap output yang dihasilkan oleh fungsi *hash* modulo, kemudian dikirim secara rahasia dan aman oleh dealer ke setiap peserta. Diasumsikan misalkan ada pihak asing yang berusaha untuk mengetahui *share* $\boldsymbol{x_\alpha}$ dari setiap peserta, namun karena penggunaan fungsi *hash* modulo pada proses perhitungannya serta pengiriman *share* $\boldsymbol{x_\alpha}$ dilakukan secara rahasia dan aman, maka pihak asing tersebut tidak akan mudah atau sulit secara matematis untuk memperolehnya.

3.2.3 Keamanan *Secret*

Misalkan seorang peserta pada subhimpunan $\boldsymbol{A_{ij}}$ berusaha untuk mendapatkan *secret* $\boldsymbol{s_i}$. Kemudian, untuk dapat membangun kembali *secret* $\boldsymbol{s_i}$, dia harus menduga *share* peserta $\boldsymbol{k}$-1 yang tidak diketahui. Sebaliknya, pihak asing, tanpa mengetahui salah satu *share*, dan hanya mengetahui bahwa panjang *secret* $\boldsymbol{s_i}$ adalah 8 bit string, juga harus menduga *share* peserta $\boldsymbol{k}$ yang tidak diketahui untuk mendapatkan *secret*. Oleh karena itu, himpunan peserta yang tidak bersesuaian tidak memiliki kemampuan lebih untuk membangun atau menemukan kembali *secret* dibandingkan

pihak asing. Oleh karena itu, skema *secret sharing* berdasarkan fungsi *hash* secara perhitungan dan penggunaan prinsip ambang (*threshold*) yang bersesuaian, serta pemilihan fungsi *hash* modulo yang sulit secara perhitungan untuk menemukan *preimage* membuat skema ini menjadi aman (Das, A. & Adhikari, A., 2011).

3.3 Sifat-sifat pada Skema *Secret Sharing* berdasarkan Fungsi *Hash*

Berdasarkan Das, A., & Adhikari, A. (2011), skema *secret sharing* berdasarkan fungsi *hash* memiliki beberapa sifat diantaranya *perfect, verifiable*, dan *ideal*.

Bersifat *perfect* karena jika *secret* bisa dibagi pada peserta dalam himpunan peserta sedemikian sehingga hanya himpunan bagian tertentu saja dari himpunan peserta tersebut yang bisa membentuk kembali *secret*, sementara himpunan bagian lainnya tidak dapat. Jadi misalkan terdapat suatu data/ pesan $\boldsymbol{M}$, kemudian $\boldsymbol{M}$ dibagi kedalam $\boldsymbol{n}$ bagian, $\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_n$ sedemikian rupa sehingga setiap peserta dari $\boldsymbol{n}$ peserta tersebut memiliki masing-masing satu buah *share*. Sebuah *secret* dapat ditemukan kembali, apabila $\boldsymbol{k}$ peserta yang bersesuaian dari $\boldsymbol{n}$ peserta bergabung bersama dengan masing-masing *share* yang mereka miliki, tetapi untuk kasus dimana hanya kurang dari $\boldsymbol{k}$ peserta yang bergabung, maka tidak akan diperoleh *secret* dari data/pesan $\boldsymbol{M}$ tersebut.

Skema *secret sharing* berdasarkan fungsi *hash* bersifat *verifiable*, karena dealer dapat memeriksa kebenaran dari hasil perhitungan fungsi *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ berdasarkan input *share* yang diberikan peserta dengan menggunakan nilai hash *pseudo share* $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ yang mana sebelumnya telah dipublikasi oleh dealer ke area umum. Disisi lainnya, peserta dapat memverifikasi kebenaran dari pesan yang diberikan dealer dengan menghitung nilai $\mathbf{H}(\mathbf{H}(\boldsymbol{M_i}))$ lalu mencocokkannya dengan nilai $\mathbf{H}(\mathbf{H}(\boldsymbol{M_i})) = \mathbf{H}(\boldsymbol{s}_i)$ yang sebelumnya telah dipublikasikan oleh dealer ke area umum.

Skema *secret sharing* bersifat *ideal* jika *secret* dan *share* memiliki ukuran panjang yang sama. Skema *secret sharing* berdasarkan fungsi *hash* ini memiliki sifat *ideal* karena baik ukuran panjang *share* setiap peserta maupun *secret* dari pesan memiliki ukuran panjang yang sama, yakni 8 bit.

Dengan sifat *perfect*, *verifiable*, *ideal*, serta efisien karena keberadaan fungsi *hash* yang memiliki kecepatan lebih secara perhitungan, dan tentu saja aman (*secure*) (Das, A. & Adhikari, A., 2011) membuat skema *secret sharing* berdasarkan fungsi *hash* menjadi salah satu solusi metode untuk pengamanan suatu informasi di jaman saat ini.

BAB 4

IMPLEMENTASI

Diilustrasikan bahwa terdapat 5 peserta Andi, Adhi, Arman, Warta, dan Arif dan 2 buah pesan rahasia. Setiap peserta dari ke- 5 peserta tersebut, masing-masing memiliki kemampuan untuk mengetahui kedua pesan tersebut, dimana diasumsikan sangat bernilai penting. Masalah muncul, ketika antara peserta satu dengan peserta lainnya tidak saling percaya, dan diinginkan suatu solusi dimana untuk orang tunggal tidak akan bisa mengetahui kedua pesan tersebut, atau dengan kata lain dibutuhkan $\boldsymbol{k}$ ($\boldsymbol{k} > 1$) peserta bersesuaian untuk dapat mengetahuinya. Setelah mereka berdiskusi, akhirnya mereka memutuskan untuk meminta dan mempercayakan kepada seorang dealer untuk dapat memecahkan masalah mereka, dan ditentukan bahwa untuk menentukan pesan rahasia diperlukan minimal 3 peserta bersama-sama. Jadi, dalam hal ini nilai $\boldsymbol{k} = 3$.

4.1 Fase Dealer

- Langkah 1 : Pemilihan

  Dealer ***D*** menentukan suatu fungsi *hash,* yakni fungsi *hash* modulo, dimana modulo yang digunakan adalah modulo dengan bilangan prima 7. Algoritma fungsi *hash* modulo mengambil input sebuah pesan dalam bentuk string dengan panjang sembarang dan menghasilkan output berupa bilangan biner dengan panjang 8 bit yang disebut dengan *message digest* dari input.

  Input berupa 2 buah pesan $\boldsymbol{M}_1, \boldsymbol{M}_2$, dimana $\boldsymbol{M}_1$ = ‘matematika’, dan $\boldsymbol{M}_2$ = ‘mipa’, yang akan menghasilkan output berupa 2 buah *secret* $\boldsymbol{s_1}, \boldsymbol{s_2}$ ($\boldsymbol{s_i}$ = **H** ($\boldsymbol{M_i}$) , untuk $\boldsymbol{i}$ =1,2) yang memiliki panjang masing-masing 8 bit. Maka langkah pertama adalah menghitung nilai *hash* dari masing-masing pesan, yaitu :

  $\boldsymbol{M_1}$ = ‘matematika’ akan menghasilkan output berupa bilangan biner, yaitu $\boldsymbol{s_1}$ = **0 0 0 0 0 1 0 0** ($\boldsymbol{s_1}$ = **H** ($\boldsymbol{M_1}$).

  $\boldsymbol{M_2}$ = ‘mipa’ akan menghasilkan output berupa bilangan biner, yaitu $\boldsymbol{s_2}$ = **0 0 0 0 0 0 1** ($\boldsymbol{s_2}$ = **H** ($\boldsymbol{M_2}$).

- Langkah 2 : Pengiriman *share*

Dealer $\boldsymbol{D}$ mengirim $\boldsymbol{x_\alpha}$ secara rahasia ke setiap peserta, dimana $\boldsymbol{x_\alpha}$ merupakan *share* untuk masing-masing peserta. Dalam contoh ini, dimisalkan bahwa input dari setiap peserta ($\boldsymbol{y_\alpha}$, untuk $\boldsymbol{\alpha}$ = 1, 2, 3, 4, 5) adalah nama masing-masing peserta. Ada 5 peserta, yakni ; Andi, Adhi, Arman, Warta, dan Arif. Kemudian dengan input berupa nama dari masing-masing peserta didapatkan nilai *share* dengan rumus $\boldsymbol{x_\alpha}$ = **H** ($\boldsymbol{y_\alpha}$). Maka akan diperoleh nilai *share* $\boldsymbol{x_1}$, $\boldsymbol{x_2}$, $\boldsymbol{x_3}$, $\boldsymbol{x_4}$, dan $\boldsymbol{x_5}$ sebagai berikut ;

untuk ‘andi’ akan memperoleh share berupa bilangan biner, yaitu
$\boldsymbol{x_1}$ = **0 0 0 0 0 1 1 0** ($\boldsymbol{x_1}$ = **H** ($\boldsymbol{y_1}$)).

untuk ‘adhi’ akan memperoleh share berupa bilangan biner, yaitu
$\boldsymbol{x_2}$ = **0 0 0 0 0 0 0 0** ($\boldsymbol{x_2}$ = **H** ($\boldsymbol{y_2}$)).

untuk ‘arman’ akan memperoleh share berupa bilangan biner, yaitu
$\boldsymbol{x_3}$ = **0 0 0 0 0 0 1 0** ($\boldsymbol{x_3}$ = **H** ($\boldsymbol{y_3}$)).

untuk ‘warta’ akan memperoleh share berupa bilangan biner, yaitu
$\boldsymbol{x_4}$ = **0 0 0 0 0 1 0 0** ($\boldsymbol{x_4}$ = **H** ($\boldsymbol{y_4}$)).

untuk ‘arif’ akan memperoleh share berupa bilangan biner, yaitu
$\boldsymbol{x_5}$ = **0 0 0 0 0 1 0 1** ($\boldsymbol{x_5}$ = **H** ($\boldsymbol{y_5}$)).

untuk menyederhanakan simbol, didefinisikan symbol $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$, $\boldsymbol{P_4}$, dan $\boldsymbol{P_5}$ sebagai berikut ;

$\boldsymbol{P_1}$; ‘Andi’

$\boldsymbol{P_2}$; ‘Adhi’

$\boldsymbol{P_3}$; ‘Arman’

$\boldsymbol{P_4}$; ‘Warta’

$\boldsymbol{P_5}$; ‘Arif’

ditentukan struktur akses yaitu berupa himpunan dan subhimpunan yang dapat mengakses pesan rahasia ;

$\Gamma \boldsymbol{s_1} = \{\boldsymbol{A_{11}}, \boldsymbol{A_{12}}, \boldsymbol{A_{13}}, \boldsymbol{A_{14}}\}$, dimana $\boldsymbol{A_{11}} = \{\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3}\}$

$\boldsymbol{A_{12}} = \{\boldsymbol{P_2}, \boldsymbol{P_3}, \boldsymbol{P_4}\}$

$\boldsymbol{A_{13}} = \{\boldsymbol{P_3}, \boldsymbol{P_4}, \boldsymbol{P_5}\}$

$\boldsymbol{A_{14}} = \{\boldsymbol{P_1}, \boldsymbol{P_3}, \boldsymbol{P_5}\}$

$\Gamma \boldsymbol{s_2} = \{\boldsymbol{A_{21}}, \boldsymbol{A_{22}}\}$, dimana $\boldsymbol{A_{21}} = \{\boldsymbol{P_1}, \boldsymbol{P_3}, \boldsymbol{P_4}\}$

$\boldsymbol{A_{22}} = \{\boldsymbol{P_2}, \boldsymbol{P_4}, \boldsymbol{P_5}\}$

dengan struktur akses tersebut, ditentukan minimal 3 peserta yang dapat mengakses pesan rahasia, akan tetapi tidak sembarang 3 peserta dapat mengakses pesan rahasia. Sebagai contoh, untuk *secret* ke-2, maka $\{\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3}\}$ tidak dapat mengakses pesan rahasia ke-2 tersebut, tetapi mereka dapat mengakses pesan rahasia ke-1.

Setelah nilai *secret* ke-$\boldsymbol{i}$ (untuk $\boldsymbol{i}$ = 1, 2) diperoleh, tahap selanjutnya dealer menghitung nilai *hash* dari *secret* ke-$\boldsymbol{i}$ untuk dipublikasikan ke area umum, dengan pehitungan ;

- Nilai hash *secret* ke-$\boldsymbol{i}$ (untuk $\boldsymbol{i}$ = 1, 2) **H** ($\boldsymbol{s_i}$).
  dimana **H** ($\boldsymbol{s_i}$) menunjukkan **H** ($\boldsymbol{s_i}$) = **H** (**H**($\boldsymbol{M_i}$))
  - $\boldsymbol{M_1}$ = ‘matematika’ menghasilkan output berupa bilangan biner, yaitu $\boldsymbol{s_1}$ = **0 0 0 0 0 1 0 0**,

    H (**H** ($\boldsymbol{M_1}$)) = **H** ($\boldsymbol{s_1}$)

    Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **0 0 0 0 0 1 0 0** harus diubah menjadi suatu string berupa ‘**00000100**’. Jadi

$\mathbf{H}(\mathbf{H}(\boldsymbol{M_1})) = \mathbf{H}(\boldsymbol{s_1})$

$= \mathbf{H}(\mathbf{00000100})$

$= \mathbf{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}$

- $\boldsymbol{M_2}$ = 'mipa' akan menghasilkan output berupa bilangan biner, yaitu $\boldsymbol{s_2}$ = **0 0 0 0 0 0 1 1**,

  $\mathbf{H}(\mathbf{H}(\boldsymbol{M_2})) = \mathbf{H}(\boldsymbol{s_2})$

  Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **0 0 0 0 0 1 0 0** harus diubah menjadi suatu string berupa '**00000011**'. Jadi
  $\mathbf{H}(\mathbf{H}(\boldsymbol{M_2})) = \mathbf{H}(\boldsymbol{s_2})$

  $= \mathbf{H}(\mathbf{00000011})$

  $=$ **0 0 0 0 0 0 0 1.**

Tabel 4.1 Tabel Hasil Perhitungan Tahap Dealer

Pesan $\boldsymbol{M_i}$ ($\boldsymbol{i}$ = 1, 2)	$\boldsymbol{s_i}$ = **H** ($\boldsymbol{M_i}$)	**H** (**H**($\boldsymbol{M_i}$))
matematika	**0 0 0 0 0 1 0 0**	**0 0 0 0 0 0 0 0**
Mipa	**0 0 0 0 0 0 1 1**	**0 0 0 0 0 0 0 1**

- Tahap pembangkit *Pseudo Share*

  Misalkan $\boldsymbol{l} = \lfloor \boldsymbol{log_2 z} \rfloor + 1$ dan $\boldsymbol{m} = \lfloor \boldsymbol{log_2 t} \rfloor + 1$, dimana $\boldsymbol{t}$ = max { $\boldsymbol{t_1}, \boldsymbol{t_2}, . . ., \boldsymbol{t_z}$} dan $\boldsymbol{t_i} = |\Gamma \boldsymbol{s_i}|$, maka

$\boldsymbol{l}$ = $\lfloor \boldsymbol{log_2} 2 \rfloor$ + 1 = 2

$\boldsymbol{t_1}$ = | $\Gamma \boldsymbol{s_1}$| = 4

$\boldsymbol{t_2}$ = | $\Gamma \boldsymbol{s_2}$| = 2

$\boldsymbol{t}$ = max { $\boldsymbol{t_1}$, $\boldsymbol{t_2}$} = 4, sehingga $\boldsymbol{m}$ = $\lfloor \boldsymbol{log_2} 4 \rfloor$ + 1 = 3

Dealer ***D*** menghitung *pseudo share* untuk masing-masing subhimpunan dari setiap struktur akses, yaitu

$\{\bigoplus_{\alpha: P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

dimana fungsi $\mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})$ akan menghasilkan suatu output untuk setiap inputnya berupa bilangan biner dengan panjang 8, yang diambil dengan *high-order*.

untuk $\boldsymbol{i}$ = 1, maka $\boldsymbol{j}$ = 1, 2, 3, 4 (karena untuk *secret* ke-1, terdapat 4 subhimpunan yang dapat merekonstruksi *secret* ke-1)

untuk $\boldsymbol{i}$ = 2, maka $\boldsymbol{j}$ = 1, 2 (karena untuk *secret* ke-2, terdapat 2 subhimpunan yang dapat merekonstruksi *secret* ke-2)

- Langkah 1 :
  - ❖ Proses perhitungan *pseudo share* untuk masing-masing subhimpunan dari setiap struktur akses, yaitu ;
    - Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 1 menunjukkan subhimpunan ke-1 dari *secret* ke-1.

      $\boldsymbol{A_{11}} = \{\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3}\}$

      $\{\bigoplus_{\alpha: P_\alpha \in A_{11}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = $\{\mathbf{F}(\boldsymbol{x_1} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_2} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_3} \| \mathbf{1}_l \| \mathbf{1}_m)\}$ dimana $\mathbf{1}_l$ {untuk $\boldsymbol{l}$ = 2} menandakan 2 bit representasi dari 1, yakni 0 1. Sedangkan $\mathbf{1}_m$ {untuk $\boldsymbol{m}$ = 3} menandakan 3 bit representasi dari 1, yakni 0 0 1.

Untuk subhimpunan $\boldsymbol{A_{11}}$, nilai *share* dari masing-masing peserta pada subhimpunan tersebut adalah

$\boldsymbol{x_1}$ = **0 0 0 0 0 1 1 0**

$\boldsymbol{x_2}$ = **0 0 0 0 0 0 0 0**

$\boldsymbol{x_3}$ = **0 0 0 0 0 0 1 0**

Sehingga,

$\{\bigoplus_{\alpha:P_\alpha \in A_{11}} \mathbf{F}(\, x_\alpha \| \, i_l \, \| j_m \,)\}$ = { **F**(0 0 0 0 0 **1 1 0**|| **01** ||**0 0 1**)
$\oplus$ **F**(0 0 0 0 0 **0 0 0**|| **0 1** ||**0 0 1**) $\oplus$
F(0 0 0 0 0 **0 1 0**||**0 1** ||**0 0 1**)}

= { **1 1 0 0 1 0 0 1** $\oplus$ **0 0 0 0 1 0 0 1** $\oplus$
0 1 0 0 1 0 0 1 }

= **1 0 0 0 1 0 0 1**

- Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 2 menunjukkan subhimpunan ke-2 dari *secret* ke-1.

$\boldsymbol{A_{12}} = \{\, \boldsymbol{P_2}, \boldsymbol{P_3}, \boldsymbol{P_4} \,\}$

$\{\bigoplus_{\alpha:P_\alpha \in A_{12}} \mathbf{F}(\, x_\alpha \| \, i_l \, \| j_m \,)\}$ = { $\mathbf{F}(\, x_2 \| \, \mathbf{1}_l \, \| \mathbf{1}_m \,) \oplus \mathbf{F}(\, x_3 \| \, \mathbf{1}_l \, \| \mathbf{1}_m \,) \oplus \mathbf{F}(\, x_4 \| \, \mathbf{1}_l \, \| \mathbf{1}_m \,)$ } dimana $\mathbf{1}_l$ {untuk $\boldsymbol{l}$ = 2} menandakan 2 bit representasi dari 1, yakni 0 1. Sedangkan $\mathbf{2}_m$ {untuk $\boldsymbol{m}$ = 3} menandakan 3 bit representasi dari 2, yakni 0 1 0.

Untuk subhimpunan $\boldsymbol{A_{12}}$, nilai *share* dari masing-masing peserta pada subhimpunan tersebut adalah

$\boldsymbol{x_2}$ = **0 0 0 0 0 0 0 0**

$\boldsymbol{x_3}$ = **0 0 0 0 0 0 1 0**

$\boldsymbol{x_4}$ = **0 0 0 0 0 1 0 0**

Sehingga,

$\{\bigoplus_{\alpha:P_\alpha \in A_{12}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ ={ **F**(0 0 0 0 0 **0 0 0**|| **01** ||**0 1 0**) ⊕

F(0 0 0 0 0 **0 1 0**||**0 1** ||**0 1 0**) ⊕
F(0 0 0 0 0 **1 0 0**|| **0 1** ||**0 1 0**) }

= {**0 0 0 0 1 0 1 0** ⊕ **0 1 0 0 1 0 1 0** ⊕ **1 0 0 0 1 0 1 0** }

= **1 1 0 0 1 0 1 0**

- Untuk i = 1, j = 3 menunjukkan subhimpunan ke-3 dari *secret* ke-1.

$A_{13} = \{P_3, P_4, P_5\}$

$\{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ = { $\mathbf{F}(x_3 \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(x_4 \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(x_5 \| \mathbf{1}_l \| \mathbf{1}_m)$ } dimana $\mathbf{1}_l$ {untuk l = 2} menandakan 2 bit representasi dari 1, yakni 0 1. Sedangkan $\mathbf{3}_m$ {untuk m = 3} menandakan 3 bit representasi dari 3, yakni 0 1 1.

Untuk subhimpunan A_{13}, nilai *share* dari masing-masing peserta pada subhimpunan tersebut adalah

x_3 = **0 0 0 0 0 0 1 0**

x_4 = **0 0 0 0 0 1 0 0**

x_5 = **0 0 0 0 0 1 0 1**

$\{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ = { **F** (0 0 0 0 0 **0 1 0**|| **01** ||**0 1 1**)
⊕ **F** (0 0 0 0 0 **1 0 0**|| **0 1** ||**0 1 1**)
⊕ **F** (0 0 0 0 0 **1 0 1**|| **0 1** ||**0 1 1**) }

= { **0 1 0 0 1 0 1 1** ⊕ **1 0 0 0 1 0 1 1** ⊕ **1 0 1 0 1 0 1 1** }

= **0 1 1 0 1 0 1 1**

- Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 4 menunjukkan subhimpunan ke-4 dari *secret* ke-1.

  $\boldsymbol{A_{14} = \{ P_1, P_3, P_5 \}}$

  $\{\bigoplus_{\alpha:P_\alpha \in A_{14}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\} = \{ \mathbf{F}(\boldsymbol{x_1} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_3} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_5} \| \mathbf{1}_l \| \mathbf{1}_m) \}$ dimana $\mathbf{1}_l$ {untuk $\boldsymbol{l}$ = 2} menandakan 2 bit representasi dari 1, yakni 0 1. Sedangkan $\mathbf{4}_m$ {untuk $\boldsymbol{m}$ = 3} menandakan 3 bit representasi dari 4, yakni 1 0 0.

  Untuk subhimpunan $\boldsymbol{A_{14}}$, nilai *share* dari masing-masing peserta pada subhimpunan tersebut adalah

  $\boldsymbol{x_1}$ = **0 0 0 0 0 1 1 0**

  $\boldsymbol{x_3}$ = **0 0 0 0 0 0 1 0**

  $\boldsymbol{x_5}$ = **0 0 0 0 0 1 0 1**

  $\{\bigoplus_{\alpha:P_\alpha \in A_{14}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ ={ **F**(0 0 0 0 0 **1 1 0**|| **01** ||**1 0 0**) ⊕ **F**(0 0 0 0 0 **0 1 0**|| **0 1** ||**1 0 0**) ⊕ **F**(0 0 0 0 0 **1 0 1**|| **0 1** ||**1 0 0**) }

  = { **1 1 0 0 1 1 0 0** ⊕ **0 1 0 0 1 1 0 0** ⊕ **1 0 1 0 1 1 0 0** }

  = **0 0 1 0 1 1 0 0**

- Untuk $\boldsymbol{i}$ = 2, $\boldsymbol{j}$ = 1 menunjukkan subhimpunan ke-1 dari *secret* ke-2.

  $\boldsymbol{A_{21} = \{ P_1, P_3, P_4 \}}$

  $\{\bigoplus_{\alpha:P_\alpha \in A_{21}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\} = \{ \mathbf{F}(\boldsymbol{x_1} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_3} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_4} \| \mathbf{1}_l \| \mathbf{1}_m) \}$ dimana $\mathbf{2}_l$ {untuk $\boldsymbol{l}$ = 2} menandakan 2 bit representasi dari 2, yakni 1 0. Sedangkan $\mathbf{1}_m$ {untuk $\boldsymbol{m}$ = 3} menandakan 3 bit representasi dari 1, yakni 0 0 1.

Untuk subhimpunan $\boldsymbol{A_{21}}$, nilai *share* dari masing-masing peserta pada subhimpunan tersebut adalah

$\boldsymbol{x_1}$ = **0 0 0 0 0 1 1 0**

$\boldsymbol{x_3}$ = **0 0 0 0 0 0 1 0**

$\boldsymbol{x_4}$ = **0 0 0 0 0 1 0 0**

Sehingga,

$\{\bigoplus_{\alpha:P_\alpha \in A_{21}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = { **F**(0 0 0 0 0 **1 1 0**|| **10** ||**0 0 1**)
⊕ **F**(0 0 0 0 0 **0 1 0**|| **1 0** ||**0 0 1**)
⊕ **F**(0 0 0 0 0 **1 0 0**|| **1 0** ||**0 0 1**) }

= { **1 1 0 1 0 0 0 1** ⊕ **0 1 0 1 0 0 0 0**
1 ⊕ **1 0 0 1 0 0 0 1** }

= **0 0 0 1 0 0 0 1**

- Untuk $\boldsymbol{i}$ = 2, $\boldsymbol{j}$ = 2 menunjukkan subhimpunan ke-2 dari *secret* ke-2.

$\boldsymbol{A_{22}} = \{ \boldsymbol{P_2}, \boldsymbol{P_4}, \boldsymbol{P_5} \}$

$\{\bigoplus_{\alpha:P_\alpha \in A_{22}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = { $\mathbf{F}(\boldsymbol{x_2} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_4} \| \mathbf{1}_l \| \mathbf{1}_m) \oplus \mathbf{F}(\boldsymbol{x_5} \| \mathbf{1}_l \| \mathbf{1}_m)$ } dimana $\mathbf{2}_l$ {untuk $\boldsymbol{l}$ = 2} menandakan 2 bit representasi dari 2, yakni 1 0. Sedangkan $\mathbf{2}_m$ {untuk $\boldsymbol{m}$ = 3} menandakan 3 bit representasi dari 2, yakni 0 1 0

Untuk subhimpunan $\boldsymbol{A_{22}}$, nilai *share* dari masing-masing peserta pada subhimpunan tersebut adalah

$\boldsymbol{x_2}$ = **0 0 0 0 0 0 0 0**

$\boldsymbol{x_4}$ = **0 0 0 0 0 1 0 0**

$\boldsymbol{x_5}$ = **0 0 0 0 0 1 0 1**

Sehingga,

$\{\bigoplus_{\alpha:P_\alpha \in A_{22}} \mathbf{F}(x_\alpha \| i_l \| j_m)\} = \{ \mathbf{F}(0\,0\,0\,0\,0\,\mathbf{0\,0\,0} \| \mathbf{10} \| \mathbf{0\,1\,0}) \oplus \mathbf{F}(0\,0\,0\,0\,0\,\mathbf{1\,0\,0} \| \mathbf{1\,0} \| \mathbf{0\,1\,0}) \oplus \mathbf{F}(0\,0\,0\,0\,0\,\mathbf{1\,0\,1} \| \mathbf{1\,0} \| \mathbf{0\,1\,0}) \}$

$= \{ \mathbf{0\,0\,0\,1\,0\,0\,1\,0} \oplus \mathbf{1\,0\,0\,1\,0\,0\,1\,0} \oplus \mathbf{1\,0\,1\,1\,0\,0\,1\,0} \}$

$= \mathbf{0\,0\,1\,1\,0\,0\,1\,0}$

Setelah nilai *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ dari subhimpunan A_{ij} (i = 1,2 ; j = 1, 2, 3, 4) diperoleh, tahap selanjutnya dealer menghitung nilai *hash* dari nilai *pseudo share* untuk dipublikasikan ke area umum, dengan perhitungan ;

- Nilai *hash pseudo share* $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ dari subhimpunan A_{ij}

  untuk i = 1, maka j = 1, 2, 3, 4

  untuk i = 2, maka j = 1, 2

  - nilai *pseudo share* untuk subhimpunan A_{11} = **1 0 0 0 1 0 0 1** ,

    nilai *hash pseudo share* untuk subhimpunan A_{11} adalah

    $\mathbf{H}(\{\bigoplus_{\alpha:P_\alpha \in A_{11}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}) = \mathbf{H}(\mathbf{1\,0\,0\,0\,1\,0\,0\,1})$

    Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **1 0 0 0 1 0 0 1** harus diubah menjadi suatu string berupa '**10001001**'. Jadi

    $\mathbf{H}(\{\bigoplus_{\alpha:P_\alpha \in A_{11}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}) = \mathbf{H}(\mathbf{10001001})$

    $= \mathbf{0\,0\,0\,0\,0\,0\,1\,0}$.

- nilai *pseudo share* untuk subhimpunan $\boldsymbol{A_{12}}$ = **1 1 0 0 1 0 1 0**,

  nilai *hash pseudo share* untuk subhimpunan $\boldsymbol{A_{12}}$ adalah
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{12}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**1 1 0 0 1 0 1 0**)

  Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **1 1 0 0 1 0 1 0** harus diubah menjadi suatu string berupa '**11001010**'. Jadi
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{12}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**11001010**)

  = **0 0 0 0 0 0 1 1**.

- nilai *pseudo share* untuk subhimpunan $\boldsymbol{A_{13}}$ = **0 1 1 0 1 0 1 1**,

  nilai *hash pseudo share* untuk subhimpunan $\boldsymbol{A_{13}}$ adalah
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{13}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**0 1 1 0 1 0 1 1**)

  Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **0 1 1 0 1 0 1 1** harus diubah menjadi suatu string berupa '**01101011**'. Jadi
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{13}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**01101011**)

  = **0 0 0 0 0 1 0 0**.

- nilai *pseudo share* untuk subhimpunan $\boldsymbol{A_{14}}$ = **0 0 1 0 1 1 0 0**,

  nilai *hash pseudo share* untuk subhimpunan $\boldsymbol{A_{14}}$ adalah
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{14}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**0 0 1 0 1 1 0 0**) ,
  Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **0 0 1 0 1 1 0 0** harus diubah menjadi suatu string berupa '**00101100**'. Jadi
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{14}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**00101100**)

  = **0 0 0 0 0 0 1 0**.

- nilai *pseudo share* untuk subhimpunan $\boldsymbol{A_{21}}$ = **0 0 0 1 0 0 0 1**,

  nilai *hash pseudo share* untuk subhimpunan $\boldsymbol{A_{21}}$ adalah
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{21}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**0 0 0 1 0 0 0 1**)

  Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **0 0 0 1 0 0 0 1** harus diubah menjadi suatu string berupa '**00010001**'. Jadi
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{21}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**00010001**)

  = **0 0 0 0 0 0 0 1**.

- nilai *pseudo share* untuk subhimpunan $\boldsymbol{A_{22}}$ = **0 0 1 1 0 0 1 0**,

  nilai *hash pseudo share* untuk subhimpunan $\boldsymbol{A_{22}}$ adalah
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{22}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**0 0 1 1 0 0 1 0**) ,
  Karena input yang digunakan dalam fungsi *hash* ini hanya berupa string dengan panjang sembarang, maka vektor **0 0 1 1 0 0 1 0** harus dirubah menjadi suatu string berupa '**00110010**'. Jadi
  H ({$\bigoplus_{\alpha:P_\alpha \in A_{22}}$ **F**($x_\alpha \| i_l \| j_m$) }) = **H** (**00110010**)

  = **0 0 0 0 0 0 1 0**.

Setelah nilai *pseudo share* {$\bigoplus_{\alpha:P_\alpha \in A_{ij}}$ **F**($x_\alpha \| i_l \| j_m$) } dari subhimpunan $\boldsymbol{A_{ij}}$ ($\boldsymbol{i}$ = 1,2 ; $\boldsymbol{j}$ = 1, 2, 3, 4) dan nilai *hash* dari nilai *pseudo share* **H** {$\bigoplus_{\alpha:P_\alpha \in A_{ij}}$ **F**($x_\alpha \| i_l \| j_m$) } diperoleh, tahap selanjutnya dealer menghitung nilai *public share* untuk dipublikasikan ke area umum, dengan perhitungan ;

❖ Nilai *public share* $\boldsymbol{S_{ij}}$ dari subhimpunan $\boldsymbol{A_{ij}}$

untuk $\boldsymbol{i}$ = 1, maka $\boldsymbol{j}$ = 1, 2, 3, 4

untuk $\boldsymbol{i}$ = 2, maka $\boldsymbol{j}$ = 1, 2

- Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 1 menunjukkan subhimpunan ke-1 untuk *secret* ke-1

  nilai *pseudo share* dari subhimpunan $\boldsymbol{A_{11}}$ $\{\bigoplus_{\alpha:P_\alpha \in A_{11}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = **1 0 0 0 1 0 0 1**, maka nilai *publik share* $\boldsymbol{S_{11}}$ adalah

  $\boldsymbol{S_{11}} = \boldsymbol{s_1} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{11}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

  = **0 0 0 0 0 1 0 0** $\oplus$ **1 0 0 0 1 0 0 1**

  = **1 0 0 0 1 1 0 1**

- Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 2 menunjukkan subhimpunan ke-2 untuk *secret* ke-1

  nilai *pseudo share* dari subhimpunan $\boldsymbol{A_{12}}$ $\{\bigoplus_{\alpha:P_\alpha \in A_{12}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = **1 1 0 0 1 0 1 0**, maka nilai *publik share* $\boldsymbol{S_{21}}$ adalah

  $\boldsymbol{S_{12}} = \boldsymbol{s_1} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{12}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

  = **0 0 0 0 0 1 0 0** $\oplus$ **1 1 0 0 1 0 1 0**

  = **1 1 0 0 1 1 1 0**

- Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 3 menunjukkan subhimpunan ke-3 untuk *secret* ke-1

  nilai *pseudo share* dari subhimpunan $\boldsymbol{A_{13}}$ $\{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = **0 1 1 0 1 0 1 1**, maka nilai *publik share* $\boldsymbol{S_{13}}$ adalah

  $\boldsymbol{S_{13}} = \boldsymbol{s_1} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

  = **0 0 0 0 0 1 0 0** $\oplus$ **0 1 1 0 1 0 1 1**

  = **0 1 1 0 1 1 1 1**

- Untuk $\boldsymbol{i}$ = 1, $\boldsymbol{j}$ = 4 menunjukkan subhimpunan ke-4 untuk *secret* ke-1

  nilai *pseudo share* dari subhimpunan $\boldsymbol{A_{14}}$ $\{\bigoplus_{\alpha:P_\alpha \in A_{14}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = **0 0 1 0 1 1 0 0**, maka nilai *publik share* $\boldsymbol{S_{14}}$ adalah

  $\boldsymbol{S_{14}} = \boldsymbol{s_1} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{14}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

  = **0 0 0 0 0 1 0 0** $\oplus$ **0 0 1 0 1 1 0 0**

  = **0 0 1 0 1 0 0 0**

- Untuk $\boldsymbol{i}$ = 2, $\boldsymbol{j}$ = 1 menunjukkan subhimpunan ke-1 untuk *secret* ke-2

  nilai *pseudo share* dari subhimpunan $\boldsymbol{A_{21}}$ $\{\bigoplus_{\alpha:P_\alpha \in A_{21}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = **0 0 0 1 0 0 0 1**, maka nilai *publik share* $\boldsymbol{S_{21}}$ adalah

  $\boldsymbol{S_{21}} = \boldsymbol{s_2} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{21}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

  = **0 0 0 0 0 0 1 1** $\oplus$ **0 0 0 1 0 0 0 1**

  = **0 0 0 1 0 0 1 0**

- Untuk $\boldsymbol{i}$ = 2, $\boldsymbol{j}$ = 2 menunjukkan subhimpunan ke-2 untuk *secret* ke-2

  nilai *pseudo share* dari subhimpunan $\boldsymbol{A_{22}}$ $\{\bigoplus_{\alpha:P_\alpha \in A_{22}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ = **0 0 1 1 0 0 1 0**, maka nilai *publik share* $\boldsymbol{S_{22}}$ adalah

  $\boldsymbol{S_{22}} = \boldsymbol{s_2} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{22}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$

  = **0 0 0 0 0 0 1 1** $\oplus$ **0 0 1 1 0 0 1 0**

  = **0 0 1 1 0 0 0 1**

Tabel 4.2 Tabel hasil perhitungan tahap pembangkit *pseudo share*

Subset	**Nilai *Pseudo Share*** $\{\bigoplus_{\alpha:P_\alpha \in A_{22}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$	**Nilai *hash Pseudo Share*** $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{22}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$	**Nilai *public share***
$\boldsymbol{A_{11}}$	**1 0 0 0 1 0 0 1**	**0 0 0 0 0 0 1 0**	**1 0 0 0 1 1 0 1**
$\boldsymbol{A_{12}}$	**1 1 0 0 1 0 1 0**	**0 0 0 0 0 0 1 1**	**1 1 0 0 1 1 1 0**
$\boldsymbol{A_{13}}$	**0 1 1 0 1 0 1 1**	**0 0 0 0 0 1 0 0**	**0 1 1 0 1 1 1 1**
$\boldsymbol{A_{14}}$	**0 0 1 0 1 1 0 0**	**0 0 0 0 0 0 1 0**	**0 0 1 0 1 0 0 0**
$\boldsymbol{A_{21}}$	**0 0 0 1 0 0 0 1**	**0 0 0 0 0 0 0 1**	**0 0 0 1 0 0 1 0**
$\boldsymbol{A_{22}}$	**0 0 1 1 0 0 1 0**	**0 0 0 0 0 0 1 0**	**0 0 1 1 0 0 0 1**

- Langkah 2 :

  Dealer ***D*** mempublikasikan ke area umum ;

  - Fungsi *hash* modulo **H**.
  - Struktur akses $\Gamma \boldsymbol{s_i}$, untuk $\boldsymbol{i}$ = 1, 2.
  - **H**($\boldsymbol{s_i}$) adalah nilai hash dari *secret* ke-$\boldsymbol{i}$, untuk $\boldsymbol{i}$ = 1, 2.
    (**H**($\boldsymbol{s_i}$) = **H**(**H** ($\boldsymbol{M_i}$))
  - **H**$\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ adalah nilai hash dari *pseudo share* untuk subhimpunan ke- $\boldsymbol{j}$, *secret* ke- $\boldsymbol{i}$.
  - Nilai *publik share* $\boldsymbol{S_{ij}}$ menandakan *public share* dari subhimpunan ke- $\boldsymbol{j}$ untuk *secret* ke- $\boldsymbol{i}$, yakni ;
    $\boldsymbol{S_{11}, S_{12}, S_{13}, S_{14}, S_{21}, S_{22}}$

- Tahap penemuan kembali *Secret*

  Misalkan subhimpunan $\boldsymbol{A_{13}}$ dari struktur akses $\Gamma \boldsymbol{s_1}$ ingin memperoleh pesan $\boldsymbol{M_1}$ .Kemudian masing-masing peserta dari subhimpunan tersebut berkumpul dan menyerahkan *share* masing-masing yang mereka miliki ke dealer.

  $\boldsymbol{A_{13}} = \{ \boldsymbol{P_3}, \boldsymbol{P_4}, \boldsymbol{P_5} \}$, dimana *share* dari ketiga peserta tersebut adalah

  $\boldsymbol{x_3}$ = **0 0 0 0 0 0 1 0**

  $\boldsymbol{x_4}$ = **0 0 0 0 0 1 0 0**

  $\boldsymbol{x_5}$ = **0 0 0 0 0 1 0 1**

  dengan menggunakan fungsi $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$, kemudian dealer menghitung untuk mendapatkan nilai *pseudo share* berdasarkan *share* yang diberikan masing-masing peserta dari subhimpunan tersebut.

  dimana $\mathbf{1}_{\boldsymbol{l}}$ {untuk $\boldsymbol{l}$ = 2} menandakan 2 bit representasi dari 1, yakni 0 1. Sedangkan $\mathbf{3}_{\boldsymbol{m}}$ {untuk $\boldsymbol{m}$ = 3} menandakan 3 bit representasi dari 3, yakni 0 1 1.

dengan menghitung $\{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$, maka diperoleh

$\{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ = { **F**(0 0 0 0 0 **0 1 0**|| **01** ||**0 1 1**) ⊕

F(0 0 0 0 0 **1 0 0**|| **0 1** ||**0 1 1**) ⊕

F(0 0 0 0 0 **1 0 1**|| **0 1** ||**0 1 1**) }

= { **0 1 0 0 1 0 1 1** ⊕ **1 0 0 0 1 0 1 1** ⊕

1 0 1 0 1 0 1 1 }

= **0 1 1 0 1 0 1 1**

Jadi nilai *pseudo share* yang diperoleh untuk subhimpunan A_{13} adalah **0 1 1 0 1 0 1 1**

Setelah nilai *pseudo share* didapatkan, kemudian dealer memverifikasi kebenaran nilai *pseudo share* yang diperoleh berdasarkan input *share* masing-masing peserta dengan nilai *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ yang mana sebelumnya dealer telah menghitung dan menyimpannya, dan juga nilai *hash pseudo share* $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ yang sebelumnya telah dipublikasi ke area umum oleh dealer.

nilai *hash pseudo share* untuk subhimpunan A_{13} yang telah dipublikasi ke area umum oleh dealer adalah **0 0 0 0 0 1 0 0**,

dealer menghitung nilai *hash* dari *pseudo share* berdasarkan input *share* yang diberikan masing-masing peserta dari subhimpunan A_{13}

$\mathbf{H}(\{\bigoplus_{\alpha:P_\alpha \in A_{11}} \mathbf{F}(x_\alpha \| i_l \| j_m)\})$ = **H** (**0 1 1 0 1 0 1 1**) = **0 0 0 0 0 1 0 0**

Karena outputnya sama dengan nilai *hash* yang sebelumnya telah dipublikasikan, dan nilai dari *pseudo share* berdasarkan input *share* yang diberikan masing-masing peserta dari subhimpunan A_{13} sama dengan nilai *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ yang mana sebelumnya dealer telah menghitung dan menyimpannya, maka nilai *pseudo share* yang diperoleh untuk subhimpunan A_{13} adalah benar.

Tahap selanjutnya dealer menghitung $s_i = S_{ij} \oplus \{\oplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$, dimana S_{ij} adalah nilai *public share* yang mana sebelumnya telah dipublikasikan ke area umum oleh dealer, dan s_i adalah nilai *hash* dari pesan ke-i ($s_i = \mathbf{H}(M_i)$).

Dimana S_{13} = **0 1 1 0 1 1 1 1**, maka

$s_1 = S_{13} \oplus \{\oplus_{\alpha:P_\alpha \in A_{13}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$

= **0 1 1 0 1 1 1 1** $\oplus$ **0 1 1 0 1 0 1 1**

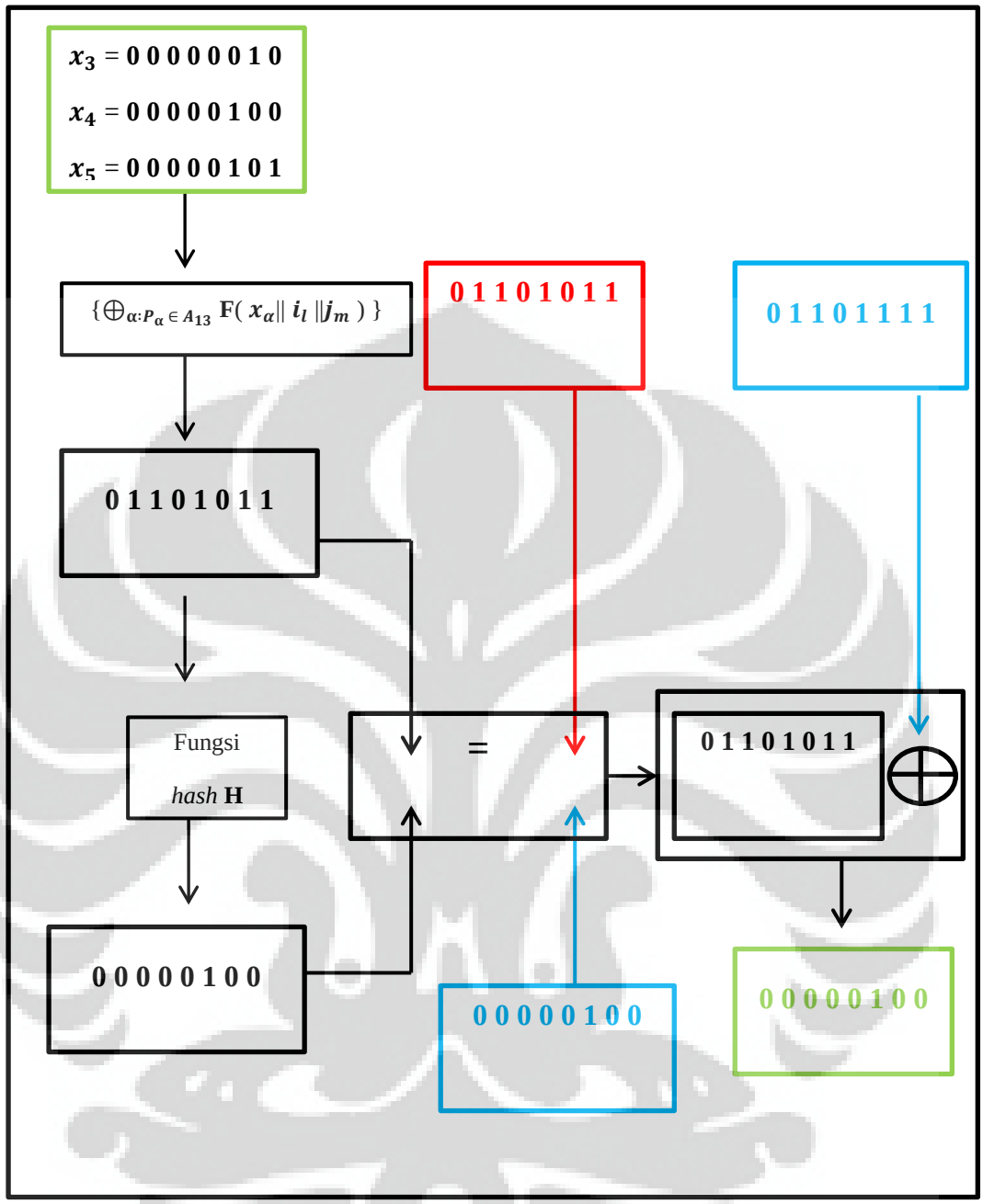
= **0 0 0 0 0 1 0 0**

Setelah nilai *hash* dari pesan ke-1 (s_1) diperoleh, kemudian dealer mencocokan nilai s_1 tersebut dengan *preimage* dari nilai tersebut yang merupakan pesan ke-1 (M_1) dimana dealer telah menyimpan sebelumnya.

Setelah dicocokan kesalah satu nilai dari *secret* s_i (i =1,2), diperoleh pesan M_1 = ‘matematika’

Setelah pesan ke-1 diketahui, kemudian dealer memberikan pesan ke-1 tersebut ke masing-masing peserta dari subhimpunan A_{13}, dan peserta dapat memverifikasi kebenaran dari pesan tersebut dengan menghitung nilai **H** (**H** (M_1)) kemudian mencocokan nilai tersebut dengan nilai **H** (**H**(M_1)) yang sebelumnya telah dipublikasikan oleh dealer ke area umum.

Dari pembahasan tentang proses perhitungan tahap penemuan kembali *secret* dari skema *secret sharing* berdasarkan fungsi *hash* untuk subhimpunan A_{13}, langkah-langkah tesebut dapat diringkas dalam ilustrasi berikut.



Gambar 3.4 Ilustrasi proses perhitungan Tahap penemuan kembali *Secret* untuk subhimpunan A_{13}

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dalam pembahasan tugas akhir ini didapat kesimpulan bahwa skema *secret sharing* tediri dari 3 komponen tahap utama, yaitu ;

1. Tahap Dealer
   - Pemilihan fungsi *hash* **H** dan himpunan peserta Γs_i untuk $i = 1,2, \ldots z$.
   - Perhitungan serta pengiriman *share* x_α untuk setiap peserta
   - Perhitungan *secret* s_i ($s_i = \mathbf{H}(M_i)$)
   - Nilai *hash* dari *secret* s_i ($\mathbf{H}(s_i) = \mathbf{H}(\mathbf{H}(M_i))$) yang mana akan dipublikasi dealer ke area umum
2. Tahap pembangkit *Pseudo Share*
   - Perhitungan *pseudo share* dengan fungsi $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$
   - Perhitungan nilai *hash* dari *pseudo share* dengan fungsi **H** $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$
   - Perhitungan *public share* dengan fungsi $S_{ij} = s_i \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$
3. Tahap penemuan kembali *Secret* s_i (*Fase Recovery*)
   - Perhitungan *secret* s_i dengan fungsi $s_i = S_{ij} \oplus \{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$, dimana S_{ij} merupakan nilai *public share* yang sebelumnya telah dipublikasi oleh dealer ke area umum, dan $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(x_\alpha \| i_l \| j_m)\}$ menyatakan nilai *pseudo share* untuk subhimpunan A_{ij}.

Berdasarkan Das, A. & Adhikari, A. (2011), skema *secret sharing* berdasarkan fungsi *hash* memiliki sifat sebagai berikut :

- *Perfect*, yaitu jika *secret* bisa didistribusikan pada himpunan peserta sedemikian sehingga hanya peserta dalam himpunan bagian tertentu saja yang bisa membangun kembali *secret*, sementara himpunan bagian lainnya tidak dapat membangun kembali *secret* (Stinson, D. R., 2002). Skema *secret sharing* berdasarkan fungsi *hash* memiliki sifat *perfect*, karena untuk dapat merekonstruksi *secret* dibutuhkan $\boldsymbol{k}$ ($\boldsymbol{k} > 1$) peserta yang bersesuaian.
- *Verifiable*, yaitu jika peserta dapat memeriksa kebenaran dari *share* mereka yang diperoleh dari dealer dan membangun kembali *secret* (Stinson, D. R., 2002). Skema ini bersifat *verifiable*, karena baik dealer maupun setiap peserta dapat memeriksa kebenaran dari nilai yang mereka telah hitung atau mereka peroleh. Dealer dapat memeriksa kebenaran dari hasil perhitungan fungsi *pseudo share* $\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ berdasarkan input *share* yang diberikan peserta dengan menggunakan nilai *hash pseudo share* $\mathbf{H}\{\bigoplus_{\alpha:P_\alpha \in A_{ij}} \mathbf{F}(\boldsymbol{x_\alpha} \| \boldsymbol{i_l} \| \boldsymbol{j_m})\}$ yang mana sebelumnya telah dipublikasi oleh dealer ke area umum. Disisi lainnya, peserta dapat memverifikasi kebenaran dari pesan yang diberikan dealer dengan menghitung nilai $\mathbf{H}(\mathbf{H}(\boldsymbol{M_i}))$ lalu mencocokkannya dengan nilai $\mathbf{H}(\mathbf{H}(\boldsymbol{M_i})) = \mathbf{H}(\boldsymbol{s_i})$ yang sebelumnya telah dipublikasikan oleh dealer ke area umum.
- *Ideal*, yaitu jika *share* dan *secret* memiliki ukuran panjang yang sama (Ernest, et al., 1991). Skema *secret sharing* berdasarkan fungsi *hash* ini memiliki sifat *ideal* karena baik ukuran panjang *share* setiap peserta maupun *secret* dari pesan memiliki ukuran panjang yang sama, yakni 8 bit.
- Skema *secret sharing* berdasarkan fungsi *hash* memiliki sifat efisien karena fungsi *hash* memiliki kecepatan lebih secara perhitungan, dan aman (*secure*) (Preneel, B., 1994).

5.2 Saran

Beberapa saran yang dapat diambil dalam tugas akhir ini, dan perlu diperhatikan adalah

1. Penggunaan fungsi modulo sebagai fungsi *hash* dalam skema ini, mempunyai beberapa kelemahan, diantaranya masih ditemukan tubrukan (*collision*), yakni untuk pesan berbeda memiliki nilai fungsi *hash* yang sama.
2. Penggunaan fungsi *hash* lainnya seperti MD 5, SHA-1, atau SHA-2, bisa menjadi alternatif untuk mengatasi masalah masih timbulnya tubrukan (*collision*).

DAFTAR PUSTAKA

Chum, C. S., & Zhang, X. (2011). *Hash Function based Secret Sharing Scheme Design*. arxiv.org/abs/1111.1278 diakses pada tanggal 12 januari 2012.

Das, A., & Adhikari, A. (2011). *An Efficient multi-use multi-secret sharing scheme based on hash function*. arxiv.org/abs/1103.1730 diakses pada tanggal 5 maret 2012.

Preneel, B. (1994). Cryptography hash function. *Proceddings of the 3rd Symposium on State and Progress of Research in Cryptography*, W. Wolfowicz (ed.), pp. 161-171, 1993. Heverlee.

Ernest, F. B., Davenport, D.M. (1991) , On the Classification of Ideal Secret Sharing Schemes. *J. Cryptology 4 (2)* : 123-134

Stallings, W. (2005). *Cryptography and Network Security Principles and Practices*. New York : Prentice Hall, vol. 4.

Rosen, K. H. (1999). *Discrete Mathematics and Its Application*. Singapore : McGraw-Hill Book, 4th ed.

Kurihara, J., Kiyamoto, S., Fukushima, K., & Tanaka, T. (2008). *A New (k, n) – Threshold Secret Sharing Scheme and Its Extension*. www.springerlink/content/x6p16295v7566203 diakses pada tanggal 3 april 2012.

Simpson, R. E. (1987). *The Exclusive OR (XOR) Gate*. dalam Allyn & Bacon, *Introductory Electronics for Scientists and Engineers vol. 2nd*. Boston. pp. 550-554.

Shamir, Adi. (1979). How to Share a Secret. *Communication of the ACM 22*, 612-613.

Stinson, D. R. (2002). *Crypthography, Theory, and Practice*. Florida : Chapman and Hall/ CRC, 2nd edition.

LAMPIRAN

Berikut ini tabel ASCII (*American Standart Code for Information Interchange*), yaitu :

Karakter	Nilai ASCII (desimal)	Karakter	Nilai ASCII (desimal)	Karakter	Nilai ASCII (desimal)	Karakter	Nilai ASCII (desimal)
!	**33**	/	**47**	=	**61**	K	**75**
"	**34**	0	**48**	>	**62**	L	**76**
#	**35**	1	**49**	?	**63**	M	**77**
$	**36**	2	**50**	@	**64**	N	**78**
%	**37**	3	**51**	A	**65**	O	**79**
&	**38**	4	**52**	B	**66**	P	**80**
‘	**39**	5	**53**	C	**67**	Q	**81**
(	**40**	6	**54**	D	**68**	R	**82**
)	**41**	7	**55**	E	**69**	S	**83**
*	**42**	8	**56**	F	**70**	T	**84**
+	**43**	9	**57**	G	**71**	U	**85**
,	**44**	:	**58**	H	**72**	V	**86**
-	**45**	;	**59**	I	**73**	W	**87**
.	**46**	<	**60**	J	**74**	X	**88**

Karakter	Nilai ASCII (desimal)	Karakter	Nilai ASCII (desimal)	Karakter	Nilai ASCII (desimal)
Y	**89**	g	**103**	t	**116**
Z	**90**	h	**104**	u	**117**
[	**91**	i	**105**	v	**118**
\	**92**	j	**106**	w	**119**
]	**93**	k	**107**	x	**120**
^	**94**	l	**108**	y	**121**
_	**95**	m	**109**	z	**122**
`	**96**	n	**110**	{	**123**
a	**97**	o	**111**	¦	**124**
b	**98**	p	**112**	}	**125**
c	**99**	q	**113**	~	**126**
d	**100**	r	**114**		
e	**101**	s	**115**		
f	**102**				

Berikut ini *Source Code* dari Algoritma Skema *Secret Sharing* berdasarkan Fungsi *Hash*.

```
clc;
fprintf ('----------skema secret sharing berdasarkan fungsi hash--------\n');
fprintf ('----------------------------------------------------------------\n');
g=input('ingin memulai menghitung skema secret sharing berdasarkan fungsi
hash (y/n) : ');
while g=='y'
clc;
fprintf ('----------skema secret sharing berdasarkan fungsi hash--------\n');
fprintf ('----------------------------------------------------------------\n');
fprintf ('menu pilihan : \n');
fprintf ('1. fase perhitungan publikshare (khusus dealer) \n');
fprintf ('2. fase perhitungan pseudoshare (khusus dealer) \n');
fprintf ('3. fase recovery (khusus dealer) \n');
fprintf ('4. fase verifikasi \n');
w = input('pilih menu yang anda inginkan berdasarkan list diatas : ');
switch w
case 1
clear;
clc;
fprintf('--------------fase perhitungan publikshare---------------------\n');
fprintf('================================================================\n');
%meminta masukkan secret dari user
fprintf('--------------meminta masukkan pesan untuk secret--------------\n');
p = input('input bilangan prima : ');
m = input('berapa banyak secret : ');
for c=1:m
pesan{c}=input(['input pesan ke-' num2str(c) ' yang ingin dibuat message
digest: ']);
bentukASCII = double(pesan{c});
[jumlahbaris,jumlahkolom]=size(bentukASCII);
   for i=2:jumlahkolom
      bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
      a=bentukASCII(i);
   i=i+1;
   end
Total=a;
nilai_modulus_p=mod(Total,p);
nilai=dec2bin(nilai_modulus_p,8);
   for b=1:length(nilai)
      k(b)=str2num(nilai(b));
   end
secret{c}=k;
secret{c};
bentukASCII = double(nilai);
[jumlahbarissecret,jumlahkolomsecret]=size(bentukASCII);
            for i=2:jumlahkolomsecret
               bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
               a=bentukASCII(i);
            i=i+1;
            end
               Total=a;
               nilai_modulus_p=mod(Total,p);
               nilai=dec2bin(nilai_modulus_p,8);
               for b=1:length(nilai)
```

```
                k(b)=str2num(nilai(b));
            end
            hashsecret{c}=k;
            hashsecret{c};
c=c+1;
end
%meminta masukkan share untuk setiap peserta
fprintf('-------------meminta masukkan share untuk setiap peserta-------\n');
g = input('berapa banyak peserta: ');
for j=1:g
   x=input(['input share untuk peserta ke-' num2str(j) ': ']);
   bentukASCII = double(x);
   [jumlahbaris,jumlahkolom]=size(bentukASCII);
   for i=2:jumlahkolom
      bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
      a=bentukASCII(i);
   i=i+1;
   end
   Total=a;
   nilai_modulus_p=mod(Total,p);
   nilai=dec2bin(nilai_modulus_p,8);
   for b=1:length(nilai)
      k(b)=str2num(nilai(b));
   end
   peserta{j}=k;
   peserta{j};
j=j+1;
end
%meminta masukkan authorized subset
fprintf('--------------meminta masukkan authorized subset---------------\n');
for c=1:m
    d{c}=input(['Jumlah subset untuk struktur access secret ke-' num2str(c)
': ']);
   for e=1:d{c}
      subset{c,e}=input(['input kombinasi yang diinginkan dalam bentuk
      matriks dari secret ke-' num2str(c) ' untuk subset ke-' num2str(e) ':
      ']);
      subset{c,e};
   e=e+1;
   end
fprintf('-----------------------------------------------------------------\n');
c=c+1;
end
clc;
%menampilkan nilai secret, share, dan nilai authorized subset yang telah
disimpan
fprintf('-nilai share dari masing-masing peserta, dan nilai authorized subset
yang telah disimpan-\n');
fprintf('-----------------------------------------------------------------\n');
fprintf('share peserta :\n');
for j=1:g
   fprintf('share peserta ke-%d yang diperoleh : \n',j);
   peserta{j}
j=j+1;
end
fprintf('=================================================================\n');
for c=1:m
```

```
      for e=1:d{c}
         fprintf('nilai kombinasi peserta dari secret ke-%d untuk subset ke-
         %d: ',c,e);
         subset{c,e}
         [jumlahbaris,jumlahkolom]=size(subset{c,e});
      e=e+1;
      end
fprintf('-----------------------------------------------------------------\n');
c=c+1;
end
fprintf('=================================================================\n');
h=input('ingin memulai proses perhitungan (y/n) : ');
while h~='y'
      h=input('ingin memulai proses perhitungan (y/n) : ');
end
clc;
%fase pseudo-share generation
l=floor(log2(m))+1;
l;
for c=1:m
   q{c}=d{c};
   c=c+1;
end
if m==1
   t=q{1};
else
   for c=2:m
      q{c}=max(q{c},q{c-1});
      t=q{c};
   c=c+1;
   end
end
s=floor(log2(t))+1;
s;
%------------------menghitung pseudo-secret share----------------------------
for c=1:m
   fprintf('untuk secret ke-%d : \n',c);
   wide=dec2bin(c,l);
   for z=1:length(wide)
      yill(z)=str2num(wide(z));
   end
   iw=yill;
   iw;
   o=length(iw);
         for e=1:d{c}
            fprintf('untuk secret ke-%d subset ke-%d : \n',c,e);
            long=dec2bin(e,s);
            for n=1:length(long)
               yull(n)=str2num(long(n));
            end
            jw=yull;
            jw;
            r=length(jw);
            [jumlahbaris,jumlahkolom]=size(subset{c,e});
                  for u=1:jumlahbaris
                     [subset{c,e}(u,:),iw,jw];
                     v=[subset{c,e}(u,:),iw,jw];
```

```
                    v(1+(o+r):8+(o+r));
                    share{c,e,u}=v(1+(o+r):8+(o+r));
                    share{c,e,u};
                    u=u+1;
                end
      e=e+1;
      end
c=c+1;
end
clc;
%perhitungan pseudo share
for c=1:m
      for e=1:d{c}
         for u=2:jumlahbaris
            share{c,e,u}=xor(share{c,e,u},share{c,e,u-1});
            u=u+1;
         end
         nilaipseudoshare{c,e}=share{c,e,u-1};
         nilaipseudoshare{c,e};
         fill=nilaipseudoshare{c,e};

         pseudoshare2{c,e}=[dec2bin(nilaipseudoshare{c,e}(1)),dec2bin(nilaips
         eudoshare{c,e}(2)),dec2bin(nilaipseudoshare{c,e}(3)),dec2bin(nilaips
         eudoshare{c,e}(4)),dec2bin(nilaipseudoshare{c,e}(5)),dec2bin(nilaips
         eudoshare{c,e}(6)),dec2bin(nilaipseudoshare{c,e}(7)),dec2bin(nilaips
         eudoshare{c,e}(8))];
         pseudoshare2{c,e};
         bentukASCII = double(pseudoshare2{c,e});
         [jumlahbarispseudoshare,jumlahkolompseudoshare]=size(bentukASCII);
            for i=2:jumlahkolompseudoshare
               bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
               a=bentukASCII(i);
            i=i+1;
            end
               Total=a;
               nilai_modulus_p=mod(Total,p);
               nilai=dec2bin(nilai_modulus_p,8);
               for b=1:length(nilai)
                  k(b)=str2num(nilai(b));
               end
               hashpseudoshare{c,e}=k;
               hashpseudoshare{c,e};
               nilaipseudoshare{c,e}=fill;
         e=e+1;
         end
c=c+1;
end
%perhitungan nilai publik
fprintf ('-----------------menampilkan nilai publik share---------------\n');
for  c=1:m
   for e=1:d{c}
      publikshare{c,e}=xor(secret{c},nilaipseudoshare{c,e});
      fprintf('nilai publik share dari secret ke-%d untuk subset ke-%d:
      ',c,e);
      pseudoshare{c,e}=[];
      publikshare{c,e}
      e=e+1;
```

```
   end
c=c+1;
end
fprintf('================================================================\n');
fprintf('-menampilkan nilai hash dari pseudo share masing-masing subset-\n');
for c=1:m
   for e=1:d{c}
      fprintf('nilai hash pseudo share dari secret ke-%d untuk subset ke-%d:
      ',c,e);
      hashpseudoshare{c,e}
   e=e+1;
   end
c=c+1;
end
fprintf('================================================================\n');
fprintf('-----------------menampilkan nilai hash dari secret------------\n');
for c=1:m
   fprintf('nilai hash dari pesan ke-%d adalah : ',c);
   hashsecret{c}
c=c+1;
end
g=input('ingin kembali ke menu utama (y/n) : ');
case 2
clc;
fprintf('---------------------fase perhitungan pseudoshare--------------\n');
fprintf('---------------------------------------------------------------\n');
%meminta masukkan authorized subset
c=input('secret keberapa yang ingin dicari: ');
e=input(['berdasarkan struktur akses secret ke-' num2str(c) ', subset
keberapakah anda : ' ]);
subset2=input('input kombinasi peserta dalam bentuk matriks : ');
[jumlahbarissubset2,jumlahkolomsubset2]=size(subset2);
if [jumlahbarissubset2,jumlahkolomsubset2]==[jumlahbaris,jumlahkolom]
%fase pseudo-share generation
l=floor(log2(m))+1;
s=floor(log2(t))+1;
dec2bin(c,l);
wide=dec2bin(c,l);
   for z=1:length(wide)
      yl(z)=str2num(wide(z));
   end
   ill=yl;
   ill;
   o=length(ill);
   long=dec2bin(e,s);
         for n=1:length(long)
           yk(n)=str2num(long(n));
         end
         jll=yk;
         jll;
         r=length(jll);
         [jumlahbaris,jumlahkolom]=size(subset2);
         for u=1:jumlahbaris
            [subset2(u,:),ill,jll];
            v=[subset2(u,:),ill,jll];
            v(1+(o+r):8+(o+r));
            share{c,e,u}=v(1+(o+r):8+(o+r));
```

```
            u=u+1;
        end
%perhitungan pseudo share
for u=2:jumlahbaris
    share{c,e,u}=xor(share{c,e,u},share{c,e,u-1});
u=u+1;
end
pseudoshare{c,e}=share{c,e,u-1};
fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d berdasarkan
input yang diberikan : \n',c,e);
pseudoshare{c,e}
fprintf('>>>>>>>>>>>>>>>>>>  nilai akan disimpan sebagai pseudoshare {%d,%d}
<<<<<<<<<<<<<<<<<<<<< \n',c,e);
g=input('ingin kembali ke menu utama (y/n) : ');
   else
    fprintf ('progam menghitung pseudo-secret share gagal karena jumlah input
    yang dimasukkan tidak sesuai \n');
   g=input('ingin kembali ke menu utama (y/n) : ');
end
case 3
clc;
fprintf('-----------------------fase recovery--------------------------\n');
fprintf('--------------------------------------------------------------\n');
c=input('secret keberapa yang ingin dicari: ');
e=input(['berdasarkan struktur akses secret ke-' num2str(c) ', subset
keberapakah anda : ' ]);
nilai2pseudoshare=pseudoshare{c,e};
nilai2pseudoshare;
if length(nilai2pseudoshare)==0
   fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d belum
    anda hitung \n',c,e);
   fprintf('--------anda tidak mendapatkan informasi apa-apa-----------\n');
   else
      fprintf('nilai publik share dari secret ke-%d untuk subset ke-%d:
      ',e,c);
      publikshare{c,e}
      fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d yang
      telah anda hitung : \n',c,e');
      pseudoshare{c,e}
      xor(publikshare{c,e},nilai2pseudoshare);
         if xor(publikshare{c,e},nilai2pseudoshare)==secret{c}
            fprintf('pesan secretnya adalah : \n');
            pesan{c}
            nilai2pseudoshare=[];
            pseudoshare{c,e}=[];
         else
            fprintf('input yang anda masukkan salah, anda tidak mendapatkan
            informasi apa-apa \n');
            nilai2pseudoshare=[];
            pseudoshare{c,e}=[];
         end
    end
    g=input('ingin kembali ke menu utama (y/n) : ');
case 4
clc;
clc;
fprintf('-----------------------fase verifikasi-----------------------\n');
```

```
fprintf('-----------------------------------------------------------------\n');
fprintf ('menu pilihan : \n');
fprintf ('1. verifikasi pseudoshare (khusus dealer) \n');
fprintf ('2. verifikasi secret \n');
f = input('pilih menu dari submenuutama-4 yang anda inginkan berdasarkan list
diatas : ');
switch f
case 1
clc;
fprintf('---------verifikasi nilai pseudoshare yang telah dihitung------\n');
c=input('secret keberapa yang ingin dicari: ');
e=input(['berdasarkan struktur akses secret ke-' num2str(c) ', subset
keberapakah anda : ' ]);
pseudoshareverifikasi=pseudoshare{c,e};
length(pseudoshareverifikasi);
if length(pseudoshareverifikasi)==0
   fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d belum
    anda hitung \n',c,e);
    else
    fprintf('nilai hash pseudo share dari secret ke-%d untuk subset ke-%d:
    ',c,e);
   hashpseudoshare{c,e}
   fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d yang
    telah anda hitung untuk diverifikasi kebenarannya :  \n',c,e);
   pseudoshare{c,e}
   fill2=pseudoshareverifikasi;
   pseudoshareverifikasi;

   pseudoshareverifikasi2=[dec2bin(pseudoshareverifikasi(1)),dec2bin(pseudosh
    areverifikasi(2)),dec2bin(pseudoshareverifikasi(3)),dec2bin(pseudosharever
    ifikasi(4)),dec2bin(pseudoshareverifikasi(5)),dec2bin(pseudoshareverifikas
    i(6)),dec2bin(pseudoshareverifikasi(7)),dec2bin(pseudoshareverifikasi(8))]
    ;
   pseudoshareverifikasi2;
   bentukASCII = double(pseudoshareverifikasi2);
   [jumlahbarispseudoshareverifikasi,jumlahkolompseudoshareverifikasi]=size(b
    entukASCII);
      for i=2:jumlahkolompseudoshareverifikasi
         bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
         a=bentukASCII(i);
      i=i+1;
      end
      Total=a;
      nilai_modulus_p=mod(Total,p);
      nilai=dec2bin(nilai_modulus_p,8);
          for z=1:length(nilai)
            y(z)=str2num(nilai(z));
         end
         hashpseudoshareverifikasi=y;
         hashpseudoshareverifikasi;
ifhashpseudoshareverifikasi==hashpseudoshare{c,e}&pseudoshareverifikasi==nila
ipseudoshare{c,e}
fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d yang anda
peroleh adalah benar \n',c,e);
pseudoshare{c,e}=fill2;
   else
```

```
   fprintf('nilai pseudo share dari secret ke-%d untuk subset ke-%d yang anda
    peroleh adalah salah \n',c,e);
   pseudoshare{c,e}=[];
   end
end

case 2
clc;
fprintf('-------verifikasi nilai secret yang telah dihitung-------------\n');
c=input('secret keberapa yang ingin dicari: ');
fprintf('nilai hash dari secret ke-%d: ',c);
hashsecret{c}
pesanverifikasi{c}=input(['input pesan yang ingin diverifikasi kebenarannya:
']);
fill3=pesanverifikasi{c};
bentukASCII = double(pesanverifikasi{c});
[jumlahbarissecretverifikasi,jumlahkolomsecretverifikasi]=size(bentukASCII);
      for i=2:jumlahkolomsecretverifikasi
         bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
         a=bentukASCII(i);
      i=i+1;
      end
      Total=a;
      nilai_modulus_p=mod(Total,p);
      nilai=dec2bin(nilai_modulus_p,8);
      for z=1:length(nilai)
         y(z)=str2num(nilai(z));
      end
      secretverifikasi{c}=y;
      secretverifikasi{c};
      secretverifikasi2{c}=[dec2bin(secretverifikasi{c}(1)),dec2bin(secretver
      ifikasi{c}(2)),dec2bin(secretverifikasi{c}(3)),dec2bin(secretverifikasi
      {c}(4)),dec2bin(secretverifikasi{c}(5)),dec2bin(secretverifikasi{c}(6))
      ,dec2bin(secretverifikasi{c}(7)),dec2bin(secretverifikasi{c}(8))];
      secretverifikasi2{c};
      bentukASCII = double(secretverifikasi2{c});
      [jumlahbarissecretverifikasi,jumlahkolomsecretverifikasi]=size(bentukAS
      CII);
         for i=2:jumlahkolomsecretverifikasi
            bentukASCII(i)=bentukASCII(i)+bentukASCII(i-1);
            a=bentukASCII(i);
         i=i+1;
         end
         Total=a;
         nilai_modulus_p=mod(Total,p);
         nilai=dec2bin(nilai_modulus_p,8);
            for z=1:length(nilai)
               yauu(z)=str2num(nilai(z));
            end
            hashsecretverifikasi{c}=yauu;
            hashsecretverifikasi{c};
if hashsecretverifikasi{c}==hashsecret{c}&secretverifikasi{c}==secret{c}
fprintf('pesan yang anda peroleh adalah benar \n',c);
pesanverifikasi{c}=fill3;
   else
   fprintf('pesan yang anda peroleh adalah salah \n',c);
   pesanverifikasi{c}=[];
```

```
   end
end
g=input('ingin kembali ke menu utama (y/n) : ');
   end
end
fprintf('------------------------Program selesai-----------------------\n');
fprintf('-------------------------terima kasih-------------------------\n');
```