

BAB II

LANDASAN TEORI

2.1 Sekilas Kriptografi

Kriptografi merupakan ilmu sekaligus seni untuk menjaga kerahasiaan berita yang ingin disampaikan ke pihak lain dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Proses untuk menyandikan sebuah pesan (Plainteks) menjadi pesan yang rahasia (cipherteks) disebut sebagai enkripsi, sedangkan proses sebaliknya disebut sebagai dekripsi [RIN06].

Pada abad ke -17, sejarah kriptografi mencatat korban ketika ratu Skotlandia, Queen Mary, dipancing setelah pesan rahasianya (pesan terenkripsi yang isinya rencana membunuh Ratu Elizabeth I) dari balik penjara pada abad pertengahan berhasil dipecahkan. Pada Perang Dunia II, pemerintah Nazi Jerman membuat mesin enkripsi yang dinamakan *Enigma*. Cipherteks yang dihasilkan oleh mesin ini berhasil dipecahkan oleh pihak Sekutu dan keberhasilan ini sering dikatakan sebagai faktor yang memperpendek Perang Dunia II.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu [RIN06]:

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain ke dalam data yang sebenarnya.
3. Otentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling

berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diotentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

4. Non-repudiasi atau nirpenyangkalan, adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

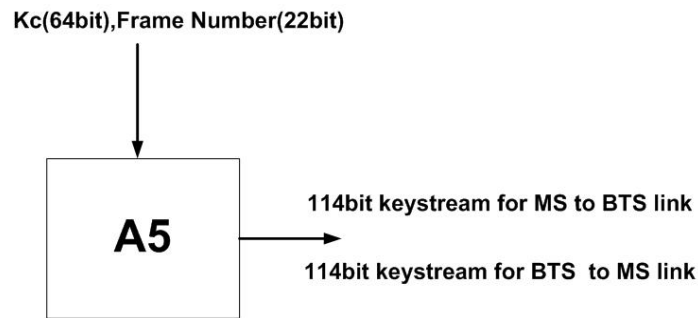
2.2 Sistem Kriptografi Kunci Simetri

Sistem kriptografi kunci-simetri merupakan sistem kriptografi yang menggunakan kunci yang sama untuk melakukan proses enkripsi dan dekripsi dari suatu pesan tertentu. Pada algoritma kriptografi modern, sistem kriptografi kunci simetri menggunakan operasi dalam mode bit yang dapat dikelompokkan menjadi *stream cipher* dan *block cipher*. *Stream cipher* beroperasi pada plainteks/cipherteks dalam bentuk bit tunggal, sedangkan *block cipher* beroperasi dalam bentuk blok-blok bit pesan. Adapun algoritma yang termasuk dalam *block cipher* dan sangat populer untuk digunakan adalah DES dan *triple* DES. Sedangkan untuk algoritma *stream cipher* yang sering digunakan adalah algoritma A5/1 untuk jaringan GSM.

DES adalah algoritma *block cipher* yang sangat populer karena dijadikan standard algoritma enkripsi kunci-simetri, meskipun saat ini standard tersebut sudah dianggap tidak aman lagi. Pada DES data dienkrip dalam blok-blok 64 bit menggunakan kunci 56 bit. DES mentransformasikan input 64 bit dalam beberapa tahap enkripsi ke dalam output 64 bit. Dengan tahapan dan kunci yang sama, DES digunakan untuk membalik enkripsi. DES sangat banyak digunakan untuk melindungi data dalam dunia elektronika khususnya di bidang perbankan, finansial, dan e-commerce. Sayangnya, DES juga mempunyai kontroversi tentang keamanannya. Penjelasan lebih detail mengenai algoritma DES terdapat pada lampiran A.1.

Algoritma A5 adalah algoritma *stream cipher* yang digunakan untuk mengenkripsi pesan dalam transmisi udara. *Stream cipher* ini diinisialisasi dengan setiap *frame* yang dikirim. *Stream cipher* ini diinisialisasi dengan kunci sesi, K_c , dan jumlah *frame* yang akan dienkripsi. Kunci sesi yang sama digunakan sepanjang panggilan berlangsung, tetapi 22 bit nomor *frame* berubah selama proses berlangsung, kemudian

membangkitkan *keystream* yang unik untuk setiap *frame* [FIR06]. Gambar II-1 menunjukkan proses pembangkitan *keystream*.



Gambar II-1 Pembangkitan *Keystream* [FIR06]

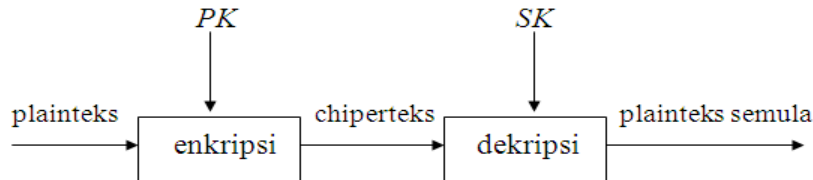
Sejak pertama kali jaringan GSM hadir, algoritma selain A5 telah didesain dan diimplementasikan. Motivasi utamanya karena algoritma enkripsi A5 yang orisinal sangat sulit untuk diterapkan di timur tengah. Sehingga algoritma A5 yang orisinal diganti namanya dengan A5/1. Algoritma lain yang termasuk di dalamnya yaitu A5/0, yang berarti tidak ada enkripsi sama sekali, dan A5/2, algoritma udara lemah. Secara umum, algoritma A5 setelah A5/1 memiliki nama A5/x. Sebagian besar algoritma A5/x lebih lemah dibandingkan dengan algoritma A5/1, yang mempunyai waktu kompleksitasnya 254. Perkiraan waktu kompleksitas A5/2 lebih rendah yaitu 216. Enkripsi ini digunakan di Amerika Serikat (USA). Penjelasan lebih detail mengenai algoritma A5/1 terdapat pada lampiran A.2.

2.3 Sistem Kriptografi Kunci-publik

Sampai akhir tahun 1975, hanya ada sistem kriptografi simetri. Karena sistem kriptografi simetri menggunakan kunci yang sama untuk enkripsi dan dekripsi, maka hal ini mengimplikasikan dua pihak yang berkomunikasi saling mempercayai. Salah satu masalah kritis dalam kriptografi kunci-simetri adalah cara mendistribusikan kunci yang digunakan.

Konsep sistem kriptografi kunci-publik ditemukan oleh Diffie dan Hellman yang mempresentasikan konsep ini pada Tahun 1976. Ide dasar dari sistem kriptografi kunci-publik adalah bahwa kunci kriptografi dibuat sepasang, satu kunci untuk

enkripsi dan satu kunci untuk dekripsi. Kunci untuk enkripsi bersifat publik (tidak rahasia) – sehingga dinamakan **kunci-publik** (*public-key*) – sedangkan kunci dekripsi bersifat rahasia – sehingga dinamakan **kunci rahasia** (*private-key* atau *secret-key*). Kunci-kunci ini dipilih sedemikian sehingga – secara praktek – tidak mungkin menurunkan kunci rahasia dari kunci-publik. Gambar II-2 menunjukkan sistem kriptografi kunci publik.



Gambar II-2 Sistem kriptografi kunci-publik [RIN06]

Sistem kriptografi kunci-publik cocok untuk kelompok pengguna di lingkungan jaringan komputer. Setiap pengguna jaringan mempunyai kunci-publik dan kunci rahasia yang bersesuaian. Kunci-publik, karena tidak rahasia, biasanya disimpan di dalam basis data kunci yang dapat diakses oleh pengguna lain. Jika ada pengguna yang hendak berkirim pesan ke pengguna lainnya, maka ia perlu mengetahui kunci-publik penerima pesan melalui basis data kunci ini lalu menggunakannya untuk mengenkripsi pesan. Hanya penerima pesan yang berhak yang dapat mendekripsi pesan karena ia mempunyai kunci rahasia. Dengan sistem kriptografi kunci-publik, tidak diperlukan pengiriman kunci rahasia melalui saluran komunikasi khusus sebagaimana pada sistem kriptografi simetri. Meskipun kunci-publik diumumkan ke setiap orang di dalam kelompok, namun kunci-publik perlu dilindungi agar otentikasinya terjamin (misalnya tidak diubah oleh orang lain).

2.3.1 Keamanan Sistem Kriptografi Kunci-Publik

Keamanan sistem kriptografi kunci-publik terletak pada dua hal [RIN06]:

1. Sulitnya menurunkan kunci rahasia dari kunci-publik.

Pada sistem kriptografi kunci-publik, kunci rahasia dan kunci-publik merupakan dua kunci yang dapat diturunkan melalui formula tertentu satu dengan yang lainnya. Tentunya formula tersebut memiliki tingkat kesukaran yang tinggi

dengan menggunakan operasi perpangkatan dan aritmatika bilangan besar sehingga sangat sulit bagi seorang kriptanalis untuk memecahkan kunci rahasia dari kunci-publik. Hal inilah yang membuat sistem kriptografi kunci-publik memiliki keamanan yang terjamin dan banyak digunakan dalam pengiriman pesan rahasia dan penting.

2. Sulitnya menurunkan plainteks dari cipherteks.

Ini merupakan pengaruh dari faktor pertama yang telah disebutkan yaitu kesukaran dalam menemukan kunci rahasia. Jika kunci rahasia suatu kriptografi kunci-publik sukar ditemukan, maka sukar pula untuk menurunkan plainteks dari cipherteks yang diterima.

Plainteks adalah pesan (teks) dari pengirim yang dikirim ke penerima melalui media tertentu, sedangkan cipherteks adalah pesan asli (plainteks) yang telah diolah (dienkripsi) dengan menggunakan algoritma kriptografi (cipher) tertentu dengan menggunakan kunci-publik.

Jika seorang kriptanalis menemukan cipherteks suatu sistem kriptografi kunci-publik dan mencoba untuk menurunkannya menjadi plainteks kembali untuk mengetahui pesan rahasia yang dikirimkan, maka ia akan menemui kesulitan yang besar karena sebelumnya kunci rahasia sistem kriptografi kunci-publik tersebut tidak dapat diturunkan dari kunci-publik yang diketahuinya. Jadi jika kunci rahasia sulit diturunkan dari kunci-publik, maka sulit pula untuk menurunkan cipherteks menjadi plainteks.

2.3.2 Kelemahan Sistem Kriptografi Kunci-publik

Kelemahan sistem kriptografi kunci-publik terletak dari waktu dan kuantitas teks yang dikirimkan. Biasanya sistem kriptografi kunci-publik menghabiskan waktu yang cukup lama untuk melakukan enkripsi dan dekripsi. Selain itu cipherteks yang dihasilkan juga lebih besar daripada plainteksnya.

Kelemahan-kelemahan sistem kriptografi kunci-publik adalah sebagai berikut [RIN06]:

1. Enkripsi dan dekripsi data umumnya lebih lambat daripada sistem simetri, karena enkripsi dan dekripsi melibatkan operasi perpangkatan yang besar.

Pengolahan plainteks, baik itu enkripsi maupun dekripsi, menggunakan algoritma kriptografi (cipher) kunci publik menggunakan operasi aritmatika dan perpangkatan yang menggunakan bilangan besar sehingga waktu untuk memproses operasi perpangkatan dan aritmatika dengan menggunakan bilangan besar tersebut memakan waktu yang lebih lama dibandingkan dengan bilangan biasa. Selain operasi perpangkatan, operasi modulo dan division yang digunakan sistem kriptografi kunci-publik yang melibatkan bilangan integer besar juga memakan waktu yang lama. Dan yang paling ekstrim adalah kombinasi operasi tersebut, yaitu operasi modulo dan perpangkatan. Kombinasi kedua operasi ini menghabiskan waktu sekitar 40 % dari proses enkripsi dan dekripsi yang dilakukan.

Sistem kriptografi kunci simetri tidak menggunakan operasi perpangkatan dengan bilangan besar seperti ini sehingga pengolahan plainteks, baik enkripsi maupun dekripsinya, lebih cepat jika dibandingkan dengan sistem kriptografi kunci-publik. Tetapi kalau dari segi keamanan sistem kriptografinya, jelas sistem kriptografi kunci-publik lebih unggul.

2. Ukuran cipherteks lebih besar daripada plainteks (bisa dua sampai empat kali ukuran plainteks)

Ukuran cipherteks yang lebih besar diakibatkan oleh adanya operasi perpangkatan dan modulo dengan menggunakan bilangan besar sehingga cipherteks yang dihasilkan memiliki ukuran yang lebih besar dibandingkan dengan plainteksnya. Ukuran cipherteks yang lebih besar ini dapat diamati dari jumlah karakter ASCII yang diperoleh pada saat enkripsi dilakukan di mana jumlah karakter ASCII tersebut lebih banyak jika dibandingkan jumlah karakter asli sebelum dilakukan proses enkripsi. Cipherteks yang memiliki ukuran yang lebih besar ini tentunya memiliki load pengiriman yang lebih besar dibandingkan dengan teks yang ukurannya lebih kecil karena lebih mudah untuk mengirimkan pesan yang ukurannya lebih kecil daripada mengirimkan pesan yang ukurannya lebih besar. Di samping itu untuk titik yang kritis, biaya untuk mengirimkan

pesan yang ukurannya lebih kecil lebih murah jika dibandingkan dengan pengiriman pesan yang ukurannya lebih besar.

Sistem kriptografi kunci simetri memiliki ukuran cipherteks sama dengan plainteksnya karena tidak menggunakan operasi perpangkatan dan modulo. Hal ini memang lebih efisien jika dibandingkan dengan sistem kriptografi kunci-publik akan tetapi segi keamanannya belum menjamin sukar dipecahkan oleh kriptanalis.

3. Karena kunci-publik diketahui secara luas dan dapat digunakan setiap orang, maka cipherteks tidak memberikan informasi mengenai otentifikasi pengirim.

Kunci-publik dapat diketahui oleh banyak orang karena hanya digunakan untuk enkripsi sedangkan kunci rahasia hanya dapat diketahui oleh seseorang yang berkaitan dengan kerahasiaan pesan. Dengan demikian, setiap orang yang mengetahui kunci-publik suatu sistem kriptografi kunci-publik tentu dapat melakukan enkripsi suatu pesan sehingga menghasilkan suatu cipherteks tertentu dan kemudian mengirimkannya ke seseorang yang memiliki kunci rahasia. Akan tetapi seseorang yang memiliki kunci rahasia tersebut tidak mengetahui siapa yang mengirimkan pesan tersebut karena cipherteks yang ia terima tidak mengandung informasi pengirim. Hal ini tentu saja tidak efektif karena bisa saja pesan yang dikirim tidak ada kaitan apa-apa mengenai urusan yang ditangani oleh seseorang yang memiliki kunci rahasia tersebut. Atau mungkin saja ada orang yang dengan iseng mengirim suatu plainteks yang isinya asal-asalan dan mengenkripsinya dengan kunci-publik suatu sistem kriptografi kunci-publik yang dimiliki instansi tertentu dan kemudian mengirimkannya ke seseorang yang memiliki kunci rahasia sementara si penerima tidak mengetahui siapa yang mengirimkannya.

Untuk sistem kriptografi kunci simetri, pihak-pihak yang menggunakannya telah mengetahui sebelumnya siapa yang memiliki kunci karena proses enkripsi dan dekripsinya menggunakan kunci yang sama. Kunci tersebut tidak *publish* ke masyarakat luas melainkan hanya diketahui oleh orang-orang yang dipercayai sehingga jika seseorang menerima cipherteks dengan sistem kriptografi kunci simetri, maka dia dapat mengetahui siapa yang mengirimkannya karena orang-orang tertentu saja yang menggunakan kunci tersebut.

2.4 Konsep Tanda-tangan Digital dengan Fungsi *Hash*

Pemberian tanda-tangan *digital* dengan cara mengenkripsinya selalu memberikan dua fungsi berbeda : kerahasiaan pesan dan otentikasi. Pada beberapa kasus, seringkali otentikasi yang diperlukan, tetapi kerahasiaan pesan tidak. Maksudnya, pesan tidak perlu dienkripsi, sebab yang dibutuhkan hanya otentikasi saja.

Hanya sistem kriptografi kunci-publik yang cocok dan alami untuk pemberian tanda-tangan *digital* dengan menggunakan fungsi *hash*. Hal ini disebabkan skema tanda-tangan *digital* berbasis sistem kunci-publik dapat menyelesaikan masalah *non-repudiation* (baik penerima dan pengirim pesan mempunyai pasangan kunci masing-masing).

2.4.1 Proses Pemberian Tanda-tangan *Digital*

Pengirim pesan mula-mula menghitung *message digest* dari pesan. *Message Digest* (*MD*) diperoleh dengan mentransformasikan pesan *M* dengan menggunakan fungsi *hash* satu arah *H*,

$$MD = H(M) \quad (\text{II-1})$$

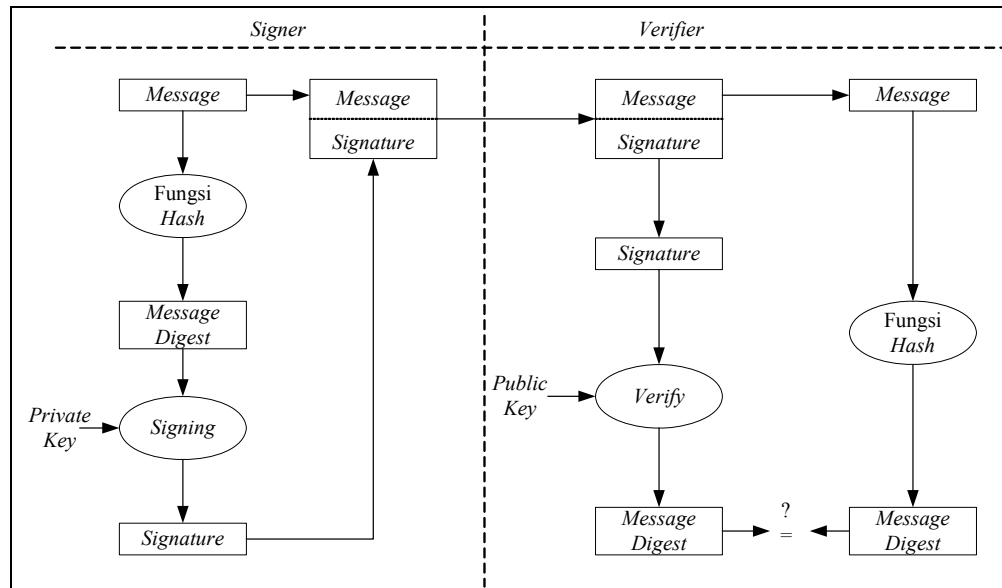
Selanjutnya, *MD* dienkripsi dengan algoritma kriptografi kunci-publik dan menggunakan kunci privat (*SK*) pengirim. Hasil enkripsi inilah yang dinamakan dengan tanda-tangan *digital S*.

Kemudian, tanda-tangan *digital S* dilekatkan ke pesan *M* (dengan cara menyambung/*append*) *S*, lalu keduanya dikirim melalui saluran komunikasi. Dalam hal ini, dapat dikatakan bahwa pesan *M* sudah ditandatangani oleh pengirim dengan tanda-tangan *digital S*. Di tempat penerima, tanda-tangan diverifikasi untuk dibuktikan keotentikannya dengan cara berikut :

1. Tanda-tangan digital *S* didekripsi dengan menggunakan kunci-publik (*PK*) pengirim pesan, menghasilkan *message digest* semula, *MD*.
2. Penerima kemudian mengubah pesan *M* menjadi *MD* menggunakan fungsi *hash* satu arah yang sama dengan fungsi *hash* yang digunakan oleh pengirim.

3. Jika $MD' = MD$, berarti tanda-tangan yang diterima otentik dan berasal dari pengirim yang benar.

Skema tanda-tangan *digital* dengan menggunakan fungsi *hash* dapat digambarkan pada Gambar II-3.



Gambar II-3 Skema Tanda Tanga Digital [RIN06]

Otentikasi pesan dapat dijelaskan sebagai berikut:

- Apabila pesan M yang diterima sudah berubah, maka MD' yang dihasilkan dari fungsi *hash* berbeda dengan MD semula. Ini berarti pesan tidak asli lagi.
- Apabila pesan M tidak berasal dari orang yang sebenarnya, maka *message digest* MD yang dihasilkan dari persamaan II-1 berbeda dengan *message digest* MD' yang dihasilkan pada proses verifikasi (hal ini karena kunci-publik yang digunakan oleh penerima pesan tidak berkoresponden dengan kunci rahasia pengirim).
- Bila $MD = MD'$, ini berarti pesan yang diterima adalah pesan yang asli (*message authentication*) dan orang yang mengirim adalah orang yang sebenarnya (*user authentication*).

2.5 Algoritma RSA

Algoritma RSA merupakan jenis algoritma kriptografi kunci-publik. Algoritma ini merupakan algoritma yang paling populer untuk digunakan. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT pada tahun 1976, yaitu : Ron(R)ivest, Adi (S)hamir, dan Leonard (A)dleman. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin. Besaran-besaran (parameter-parameter) yang digunakan algoritma RSA [RIN06]:

1. Bilangan prima, p dan q (rahasia)
2. $n = p \cdot q$ (tidak rahasia)
3. $\Phi(n) = (p - 1)(q - 1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia)

2.5.1 Perumusan Algoritma RSA

Algoritma RSA didasarkan pada teorema Euler. Persamaan teorema Euler terletak pada persamaan II-2.

$$a^{\Phi(n)} \equiv 1 \pmod{n} \quad (\text{II-2})$$

dengan persyaratan sebagai berikut :

1. a relatif prima terhadap n
2. $\Phi(n) = n(1 - 1/p_1)(1 - 1/p_2)\dots(1 - 1/p_r)$, dalam hal ini p_1, p_2, \dots, p_r adalah faktor prima dari n .

Berdasarkan sifat $ak \equiv bk \pmod{n}$ untuk k bilangan bulat ≥ 1 , maka persamaan dapat ditulis menjadi persamaan II-3.

$$a^{k\Phi(n)} \equiv 1 \pmod{n} \quad (\text{II-3})$$

Bila a diganti dengan m , maka persamaan menjadi $m^{k\Phi(n)} \equiv 1 \pmod{n}$. Berdasarkan sifat $ac \equiv bc \pmod{n}$, maka didapatkan persamaan II-4.

$$m^{k\Phi(n)+1} \equiv m \pmod{n} \quad (\text{II-4})$$

m relatif prima terhadap n .

Misalkan e dan d dipilih sedemikian sehingga

$$e \cdot d \equiv 1 \pmod{\Phi(n)} \quad (\text{II-5})$$

atau

$$e \cdot d = k \Phi(n) + 1 \quad (\text{II-6})$$

Dengan substitusi didapatkan :

$m^{e \cdot d} \equiv m \pmod{n}$, yang artinya perpangkatan m dengan e diikuti dengan perpangkatan dengan d menghasilkan kembali m semula. Berdasarkan persamaan, maka enkripsi dan dekripsi dirumuskan sebagai persamaan II-7 dan II-8.

$$E_e(m) = c \equiv m^e \pmod{n} \quad (\text{II-7})$$

$$D_d(c) = m \equiv c^d \pmod{n} \quad (\text{II-8})$$

2.5.2 Algoritma Membangkitkan Pasangan Kunci

Untuk membangkitkan sepasang kunci pada algoritma RSA, digunakan langkah-langkah sebagai berikut :

1. Pilih dua buah bilangan prima sembarang, p dan q .
2. Hitung $n = p \cdot q$
3. Hitung $\Phi(n) = (p - 1)(q - 1)$
4. Pilih kunci-publik, e , yang relatif prima terhadap $\Phi(n)$.

5. Bangkitkan kunci privat dengan persamaan $e \cdot d = k \Phi(n) + 1$, sehingga secara sederhana d dapat dihitung dengan

$$d = (1 + k \Phi(n)) / e \quad (\text{II-9})$$

Hasil dari algoritma persamaan II-9 adalah :

- Kunci-publik adalah pasangan (e, n)
- Kunci privat adalah pasangan (d, n)

n tidak bersifat rahasia, sebab ia diperlukan pada perhitungan enkripsi / dekripsi.

2.5.3 Enkripsi

Prosedur enkripsi dalam algoritma RSA adalah sebagai berikut:

1. Ambil kunci-privat penerima pesan, d , dan modulus n .
2. Nyatakan plaintext m menjadi blok-blok m_1, m_2, \dots , sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang $[0, n - 1]$.
3. Setiap blok m , dienkripsi menjadi blok c_i dengan rumus : $c_i = m_i^d \bmod n$.

2.5.4 Dekripsi

Untuk mendekripsi blok ciphertexts menjadi blok plaintexts m dilakukan dengan rumus : $c_i = m_i^e \bmod n$. Misalkan plaintext yang akan dienkripsikan adalah

$X = \text{HARI INI}$

atau dalam sistem desimal (pengkodean ASCII) adalah

7265827332737873

Pecah X menjadi blok yang lebih kecil, misalnya X dipecah menjadi enam blok yang berukuran 3 digit:

$$x_1 = 726 \qquad x_4 = 273$$

$$x_2 = 582 \qquad x_5 = 787$$

$$x_3 = 733 \qquad x_6 = 003$$

Nilai-nilai x_i ini masih terletak di dalam rentang 0 sampai 3337 – 1 (agar transformasi menjadi satu-ke-satu). Blok-blok plainteks dienkripsikan sebagai berikut:

$$726^{79} \bmod 3337 = 215 = y_1$$

$$582^{79} \bmod 3337 = 776 = y_2$$

$$733^{79} \bmod 3337 = 1743 = y_3$$

$$273^{79} \bmod 3337 = 933 = y_4$$

$$787^{79} \bmod 3337 = 1731 = y_5$$

$$003^{79} \bmod 3337 = 158 = y_6$$

Jadi, cipherteks yang dihasilkan adalah

$$Y = 215\ 776\ 1743\ 933\ 1731\ 158.$$

Dekripsi dilakukan dengan menggunakan kunci publik

$$SK = 1019$$

Blok-blok cipherteks didekripsikan sebagai berikut:

$$215^{1019} \bmod 3337 = 726 = x_1$$

$$776^{1019} \bmod 3337 = 582 = x_2$$

$$1743^{1019} \bmod 3337 = 733 = x_3$$

...

Blok plainteks yang lain dikembalikan dengan cara yang serupa. Akhirnya diperoleh kembali plainteks semula, yaitu :

$$P = 7265827332737873$$

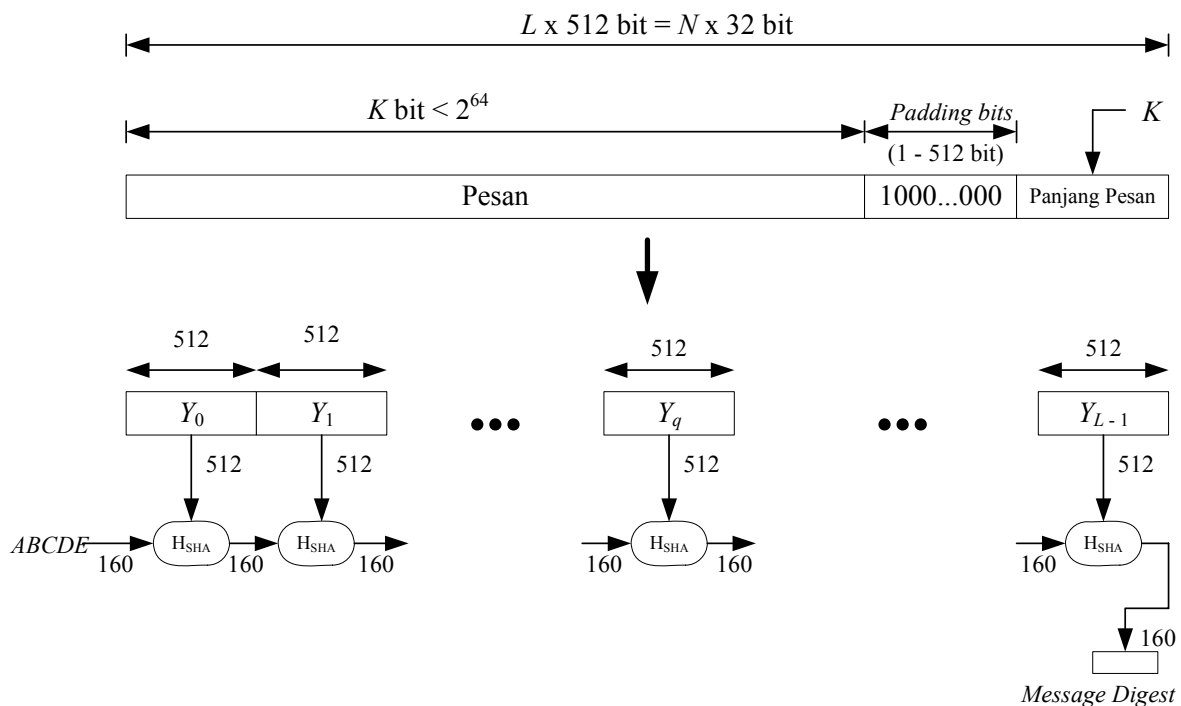
yang dalam karakter ASCII adalah

$$P = \text{HARI INI.}$$

2.6 Secure Hash Algorithm (SHA)

SHA adalah fungsi hash satu-arah yang dibuat oleh NIST dan digunakan bersama DSS (*Digital Signature Standard*). Oleh NSA, SHA dinyatakan sebagai standard fungsi *hash* satu-arah. SHA didasarkan pada MD4 yang dibuat oleh Ronald L. Rivest dari MIT. SHA disebut aman (*secure*) karena ia dirancang sedemikian sehingga secara komputasi tidak mungkin menemukan pesan yang berkoresponden dengan *message digest* yang diberikan.

Algoritma SHA menerima masukan berupa pesan dengan ukuran maksimum 2^{64} bit (2.147.483.648 gigabyte) dan menghasilkan *message digest* yang panjangnya 160 bit, lebih panjang dari *message digest* yang dihasilkan oleh MD5. Walaupun *message digest* (nilai *hash*) yang dihasilkan lebih besar ukurannya, SHA memberikan jaminan keamanan dalam hal integritas data (*data integrity*) yang lebih baik dibandingkan dengan MD5. Gambaran pembuatan *message digest* dengan algoritma SHA diperlihatkan pada Gambar II-4.



Gambar II-4 Pembuatan *Message Digest* dengan algoritma SHA [RIN06]

Langkah-langkah pembuatan *message digest* secara garis besar adalah sebagai berikut:

1. Penambahan bit-bit pengganjal (*padding bits*)
2. Penambahan nilai panjang pesan semula.
3. Inisialisasi penyangga (*buffer*) MD.
4. Pengolahan pesan dalam blok berukuran 512 bit.

2.6.1 Penambahan Bit-Bit Pengganjal

Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512. Ini berarti panjang pesan setelah ditambah bit-bit pengganjal adalah 64 bit kurang dari kelipatan 512. Angka 512 ini muncul karena SHA memproses pesan dalam blok-blok yang berukuran 512.

Pesan dengan panjang 448 bit pun tetap ditambah dengan bit-bit pengganjal. Jika panjang pesan 448 bit, maka pesan tersebut ditambah dengan 512 bit menjadi 960 bit. Jadi, panjang bit-bit pengganjal adalah antara 1 sampai 512. Bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0. Panjangnya seperti yang telah dijelaskan adalah antara 1 sampai 512.

2.6.2 Penambahan Nilai Panjang Pesan Semula

Pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Panjang pesan semula merupakan ukuran pesan asli sebelum ditambahkan dengan bit-bit pengganjal. Ukuran pesan semula ini dinyatakan dengan ukuran *byte*. Setelah ditambah dengan 64 bit, panjang pesan selanjutnya menjadi 512 bit.

2.6.3 Inisialisasi Penyangga MD

SHA membutuhkan 5 buah penyangga (*buffer*) yang masing-masing penjangnya 32 bit (MD5 hanya mempunyai 4 buah penyangga). Total panjang penyangga adalah $5 \times 32 = 160$ bit. Keempat penyangga ini menampung hasil antara dan hasil akhir.

Kelima penyangga ini diberi nama A, B, C, D, dan E. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

A = 67452301

B = EFCDAB89

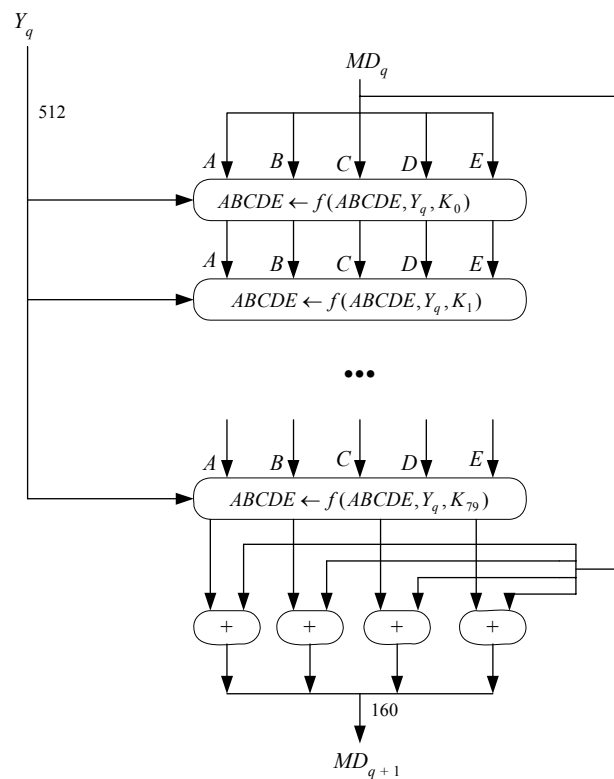
C = 98BADCFE

D = 10325476

E = C3D2E1F0

2.6.4 Pengolahan Pesan dalam Blok Berukuran 512 Bit

Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y_0 sampai Y_{L-1}). Setiap blok 512-bit diproses bersama dengan penyangga MD menjadi keluaran 160-bit, dan ini disebut proses H_{SHA} . Pengolahan pesan dalam blok berukuran 512 bit ini banyak menggunakan fungsi-fungsi logika dengan kombinasi yang berbeda pada tiap-tiap tahap putaran. Hasil pengolahan pada suatu blok pesan 512 bit menjadi masukan bagi blok pesan 512 bit berikutnya. Demikian seterusnya sampai pengolahan blok ini sampai pada blok pesan 512 bit yang terakhir seperti pada Gambar II-5.



Gambar II-5 Pengolahan blok 512 bit (Proses HSHA) [RIN06]

Proses H_{SHA} terdiri dari 80 buah putaran (MD5 hanya 4 putaran), dan masing-masing menggunakan bilangan penambah K , yaitu:

Putaran $0 \leq t \leq 19$ $K_t = 5A827999$

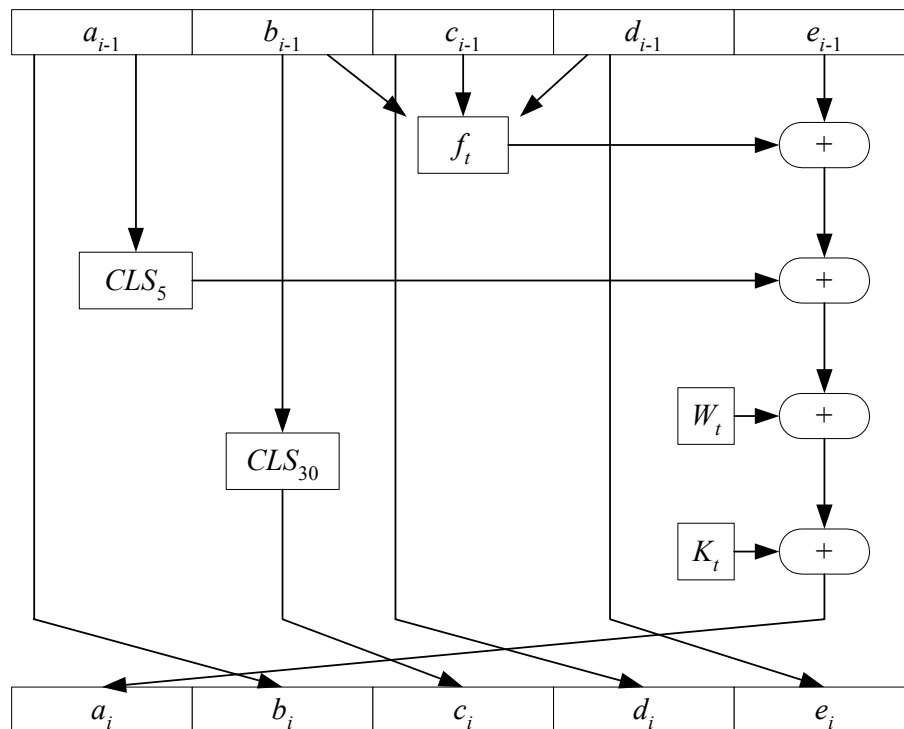
Putaran $20 \leq t \leq 39$ $K_t = 6ED9EBA1$

Putaran $40 \leq t \leq 59$ $K_t = 8F1BBCDC$

Putaran $60 \leq t \leq 79$ $K_t = CA62C1D6$

Pada Gambar II-5, Y_q menyatakan blok 512-bit ke- q dari pesan yang telah ditambah bit-bit pengganjal dan tambahan 64 bit nilai panjang pesan semula. MD_q adalah nilai *message digest* 160-bit dari proses H_{SHA} ke- q . Pada awal proses, MD_q berisi nilai inisialisasi penyangga MD.

Setiap putaran menggunakan operasi dasar yang sama (dinyatakan sebagai fungsi f). Operasi dasar SHA diperlihatkan pada Gambar II-6.



Gambar II-6 Operasi dasar SHA dalam satu putaran (fungsi f) [RIN06]

Operasi dasar SHA yang diperlihatkan pada Gambar II-6 dapat ditulis dengan persamaan II-10.

$$a, b, c, d, e \leftarrow (\text{CLS}_5(a) + f_t(b, c, d) + e + W_t + K_t), a, \text{CLS}_{30}(b), c, d \quad (\text{II-10})$$

yang dalam hal ini,

a, b, c, d, e = lima buah peubah penyangga 32-bit (berisi nilai penyangga A, B, C, D, E)

t = putaran $0 \leq t \leq 79$

f_t = fungsi logika

CLS_s = *circular left shift* sebanyak s bit

W_t = *word* 32-bit yang diturunkan dari blok 512 bit yang sedang diproses

K_t = konstanta penambah

$+$ = operasi penambahan modulo 2^{32}

atau dapat dinyatakan dalam kode program berikut:

```

for t ← 0 to 79 do
    TEMP ← (a <<< 5) + ft(b, c, d) + e + Wt + Kt
    e ← d
    d ← c
    c ← b <<< 30
    b ← a
    a ← TEMP
endfor

```

Dalam hal ini, <<< menyatakan operasi pergeseran *circular left shift*. Fungsi f_t adalah fungsi logika yang melakukan operasi logika bitwise. Operasi logika yang dilakukan dapat dilihat pada Tabel II-1.

Tabel II-1 Fungsi logika f_t pada setiap putaran [RIN06]

Putaran	$f_t(b, c, d)$
0 .. 19	(b and c) or (~b and d)

Putaran	$f_t(\mathbf{b}, \mathbf{c}, \mathbf{d})$
20 .. 39	$\mathbf{b} \text{ xor } \mathbf{c} \text{ xor } \mathbf{d}$
40 .. 59	$(\mathbf{b} \text{ and } \mathbf{c}) \text{ or } (\mathbf{b} \text{ and } \mathbf{d}) \text{ or } (\mathbf{c} \text{ and } \mathbf{d})$
60 .. 79	$\mathbf{b} \text{ xor } \mathbf{c} \text{ xor } \mathbf{d}$

Nilai W_1 sampai W_{16} berasal dari 16 word pada blok yang sedang diproses, sedangkan nilai W_t berikutnya didapatkan dari persamaan II-11.

$$W_t = W_{t-16} \text{ xor } W_{t-14} \text{ xor } W_{t-8} \text{ xor } W_{t-3} \quad (\text{II-11})$$

Setelah putaran ke-79, a , b , c , d , dan e ditambahkan ke A , B , C , D , dan E dan selanjutnya algoritma memproses untuk blok data berikutnya (Y_{q+1}). Keluaran akhir dari algoritma SHA adalah hasil penyambungan bit-bit di A , B , C , D , dan E .

2.7 Short Message Service (SMS)

Short Message Service (SMS) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi kasi tanpa kabel, memungkinkan dialkukannya pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan atau antara terminal pelanggan dengan sistem eksternal seperti email, *paging*, *voice mail*, dan lain-lain [ROM04]. Isu SMS pertama kali muncul di belahan Eropa pada sekitar tahun 1991 bersama dengan sebuah teknologi komunikasi *wireless* yang saat ini cukup banyak penggunaannya, yaitu *Global System for Mobile Communication* (GSM). Dipercaya bahwa pesan pertama yang dikirimkan menggunakan SMS dilakukan pada bulan Desember 1992, dikirimkan dari sebuah Personal Computer (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik Vodafone Inggris. Perkembangannya kemudian merambah ke benua Amerika, dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti BellSouth Mobility, PrimeCo, Nextel, dan beberapa operator lain. Teknologi digital yang digunakan bervariasi dari yang berbasis GSM, *Time Division Multiple Access* (TDMA), hingga *Code Division Multiply Access* (CDMA).

Tidak diragukan lagi SMS sangat sukses di pasaran, di tempat kelahirannya sendiri, yaitu Eropa, trafik SMS mencapai lebih dari 3 miliar per bulan meskipun tanpa ada program marketing yang proaktif dari operator seluler dan vendor pembuat perangkat komunikasi bergerak. Kesuksesan SMS dianggap sebagai kesuksesan yang tidak disengaja dan cukup mengejutkan bagi pihak-pihak yang terjun dalam industri telekomunikasi bergerak karena beberapa pihak yang berkompeten sebelumnya memprediksi bahwa SMS tidak akan laku karena penggunaanya cukup sulit dan materi untuk marketingnya sulit ditentukan.

SMS menjadi fenomena tersendiri, dalam waktu yang cukup singkat pertumbuhannya sangat tinggi tanpa ada penurunan tarif yang berarti, bahkan dapat dikatakan tarifnya mengambil posisi *steady state*. Biasanya, bahkan dalam kasus layanan telepon bergerak, tarif akan turun seiring dengan meningkatnya pengguna. Fakta lainnya adalah fasilitas SMS dalam telepon bergerak ternyata punya andil cukup besar dalam menarik kaum muda masuk dalam pasar telepon bergerak.

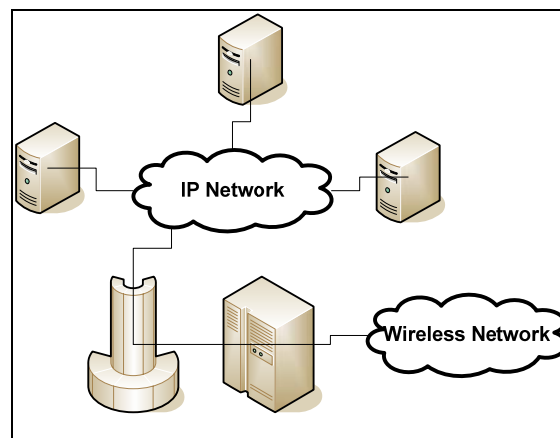
Dalam sistem SMS, mekanisme utama yang dilakukan dalam sistem adalah melakukan pengiriman pesan singkat dari satu terminal pelanggan ke terminal yang lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Center (SMSC)*, disebut juga dengan *Message Center (MC)*. SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward* trafik SMS. Di dalamnya termasuk penentuan atau pencarian rute tujuan akhir dari pesan SMS. Sebuah SMSC biasanya didesain untuk dapat menangani SMS dari berbagai sumber seperti *Voice Mail System (VMS)*, *Web-based messaging*, *Email Integration*, *External Short Message Entity (ESME)*, dan lain-lain. Dalam interkoneksi dengan entitas dalam jaringan komunikasi *wireless* seperti *Home Location Register (HLR)* dan *Mobile Switching Center (MSC)*, SMSC biasanya selalu menggunakan *Signal Transfer Point (STP)*.

Layanan SMS merupakan sebuah layanan yang bersifat *nonreal time* dimana sebuah pesan SMS dapat di-*submit* ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu pesan SMS hingga sampai ke tujuan. Kegagalan pengiriman yang bersifat sementara seperti tujuan tidak aktif akan selalu

teridentifikasi sehingga pengiriman ulang pesan SMS akan selalu dilakukan kecuali bila diberlakukan aturan bahwa pesan SMS yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

Karakteristik utama SMS adalah SMS merupakan sebuah sistem pengiriman data yang berifat *out-of-band* dengan *bandwidth* kecil. Dengan karakteristik ini, pengiriman suatu *burst data* yang pendek dapat dilakukan dengan efisiensi yang sangat tinggi. Pada awalnya SMS diciptakan untuk menggantikan layanan *paging* dengan menyediakan layanan serupa yang bersifat *two-way messaging* ditambah dengan *notification service*, khususnya untuk *voice mail*. Pada perkembangan selanjutnya, muncul jenis-jenis layanan lain seperti email, fax, dan *integration*, *interactive banking*, *information service*, dan integrasi dengan aplikasi berbasis internet. Selain itu juga berkembang layanan data *wireless* seperti *SIM download for activation*, *debet*, *profile editing*, dan lain-lain yang kemudian mendorong timbulnya layanan-layanan seperti *web-based messaging*, *gaming*, dan *chatting*.

Layanan SMS dibangun dari berbagai entitas yang saling terkait dan mempunyai fungsi dan tugas masing-masing. Tidak ada satupun dalam sistem SMS yang dapat bekerja secara parsial. Entitas dalam jaringan SMS ini disebut juga elemen jaringan SMS. Secara umum arsitektur sistem SMS, khususnya untuk sistem yang diintegrasikan dengan jaringan *wireless* terdapat pada Gambar II-7.



Gambar II-7 Arsitektur jaringan SMS

2.8 SMS-Banking

SMS Banking merupakan suatu layanan bank yang memudahkan pelanggan untuk melakukan transaksi perbankan hanya dengan menggunakan perangkat seluler mereka. Transaksi tersebut dilakukan melalui SMS yang dikirimkan secara langsung ke nomor tujuan bank, atau dapat juga terimplementasi dalam *sim card* telepon seluler pelanggan. Aplikasi yang tertanam pada *sim card* telepon seluler ini menyimpan beberapa informasi mengenai transaksi yang bisa dilakukan dengan menggunakan tarif SMS.

Berikut beberapa layanan yang disediakan oleh Bank untuk dapat melakukan transaksi melalui SMS :

- cek saldo
- cek kurs valuta asing
- cek tiga transaksi terakhir
- cek tagihan mitra
- pembayaran tagihan mitra
- pembayaran kartu kredit
- transfer antar rekening

layanan ini menjanjikan mobilitas yang tinggi, bisa dilakukan kapanpun dan dimanapun, bahkan saat roaming internasional sekalipun, tentu saja dengan syarat adalah sistem yang digunakan GSM dan nomor ponsel sudah terdaftar untuk roaming internasional.

2.8.1 Keamanan SMS-Banking

Keamanan menjadi faktor yang sangat penting untuk transaksi menggunakan *SMS-Banking*. Pada umumnya sistem keamanan yang berada pada aplikasi *SMS-Banking* menggunakan algoritma DES dan 3DES. Keamanan ini digunakan dengan menanamkan kunci pada *SIM card* telepon seluler pengguna. Untuk selanjutnya setiap transaksi *SMS-Banking* yang melalui SMS akan dienkripsi terlebih dahulu sehingga pesan transaksi tersembunyi.

Selain tertanam pada *SIM card*, jenis aplikasi *SMS-Banking* yang lain selama ini tanpa menggunakan enkripsi pesan SMS. Aplikasi tersebut mengandalkan keamanan

yang telah disediakan oleh jaringan GSM operator yaitu dengan menggunakan algoritma *stream cipher* A5/1.

Sampai dengan saat ini, masih belum terdapat jenis keamanan *SMS-Banking* yang menggunakan tanda-tangan *digital* sebagai keamanan transaksi perbankan secara *mobile*. Dengan kata lain, proses otentikasi dan autentikasi pesan transaksi belum cukup terjaga keamanannya.