Fermat's Little Theorem dan Aplikasinya pada Algoritma RSA

Akbar Gumbira - 13508106

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if18106@students.if.itb.ac.id

ABSTRAK

Pada makalah ini, penulis membahas tentang Fermat's Little Theorem beserta beberapa pembuktiannya dari beberapa sudut pandang. Selain itu penulis juga menyertakan aplikasi Fermat's Little Theorem pada Algoritma RSA. Fermat's Little Theorem ini merupakan teorema yang mendasar dari ranah ilmu teori bilangan. Bahkan dengan menggunakan teorema ini, kita dapat menurunkan Euler's Theorem dengan bantuan sifat dari fungsi Euler φ, walaupun sebenarnya Fermat's Little Theorem ini merupakan kasus khusus dari Euler's Theorem. Kemudian penulis membahas Algoritma RSA serta memberikan bukti bahwa proses dekripsi RSA valid dengan menggunakan Fermat's Little Theorem. Keamanan dari Algoritma RSA ini akan terjamin selama belum ada algoritma yang efisien untuk memfaktorkan bilangan komposit menjadi faktor primanya.

Kata kunci: Fermat's Little Theorem, RSA, prima.

1. PENDAHULUAN

Pada ranah ilmu bilangan, hipotesis China merupakan konjektur yang terbantahkan. Hipotesis ini menyatakan bahwa sebuah bilangan bulat n merupakan bilangan prima, jika dan hanya jika memenuhi kondisi $n|2^n - 2$. Atau dalam kata lain, n merupakan bilangan prima jika dan hanya jika $2^n \equiv 2\ (mod\ n)$. Hipotesis ini berlaku jika n bilangan prima, maka $2^n \equiv 2\ (mod\ n)$, yang merupakan kasus khusus dari Fermat's Little Theorem. Namun, konversnya dari hipotesis ini, dalam hal ini jika $2^n \equiv 2\ (mod\ n)$ maka n adalah bilangan prima, adalah salah. Oleh karena itu, hipotesis China tersebut secara keseluruhan adalah salah. Contoh penyangkal terkecil dari hipotesis ini adalah 341. Kemudian Fermat menggeneralisasi hipotesis tersebut yang sekarang dikenal dengan Fermat's Little Theorem. Menurut sejarah teorema ini diberikannya tanpa pemberian bukti darinya, karena menurutnya bukti yang dimilikinya terlalu panjang. Fermat's Little Theorem ini tertulis pada sebuah surat yang dikirim oleh Fermat untuk temannya, yaitu Frencle De Bessy. Setelah itu, Gottfried Leibniz pada sebuah manuskrip tanpa tanggal memberikan bukti dari teorema tersebut dan dia juga menulis bahwa dia telah mengetahui bukti tersebut sebelum tahun 1683. Namun, konvers dari generalisasi Fermat's Little Theorem tersebut juga tidak berlaku. Bilangan-bilangan komposit yang memenuhi teorema uji keprimaan dengan menggunakan Fermat's Little Theorem ini dinamakan fermat *pseudoprimes.*

Selain digunakan untuk menguji keprimaan dari suatu bilangan, Fermat's Little Theorem juga digunakan sebagai dasar dari Algoritma RSA.

RSA merupakan algoritma untuk mengenkripsi kunci publik, hal ini berarti instruksi untuk mengenkripsi sebuah pesan mungkin diketahui oleh umum walaupun algoritma dalam mendeskripsi pesan tesebut aman. Algoritma ini dinamakan dari penemunya, yaitu Ron Rivest, Adi Shamir, dan Len Adleman, yang pertama kali mengumumkannya pada tahun 1977 ketika bekerja di MIT. RSA merupakan algoritma pertama yang cocok untuk *digital signature*. RSA sampai saat ini masih digunakan secara luas dalam protokol *electronic commerce.*

2. Fermat's Little Theorem

Pierre De Fermat lahir di Beaumont-de-Lomagne, Tarn-et-Garonne, Prancis. Ia memberikan banyak sekali kontribusi pada ilmu teori bilangan. Salah satu teoremanya yang terkenal adalah Fermat Little Theorem. Teorema ini pertama kali dinyatakannya pada sebuah surat untuk temannya, Frencle de Bessy, pada tanggal 18 Oktober 1640. Pada surat tersebut tertulis : p membagi a^{p-1}-1 untuk p suatu bilangan prima dan a saling prima dengan p.

Secara formal, Fermat's Little Theorem ini dapat ditulis :
Misalkan a suatu bilangan bulat positif dan p suatu bilangan prima. Maka berlaku $a^p \equiv a\ (mod\ p)$.
Untuk $GCD(a, p) = 1$, berlaku $a^{p-1} \equiv 1\ (mod\ p)$.

2.1 Pembuktian Fermat's Little Theorem

Pada makalah ini, penulis memberikan ulasan tentang bukti dari Fermat's Little Theorem ini, bukti dari Fermat's Little Theorem ini adalah sebagai berikut :

- **Pembuktian Menggunakan Induksi**
  Sebagai basis induksi, untuk $a = 1$, teorema ini valid, sebab $p|1^p - 1$.
  Kemudian untuk langkah induksi, misalkan teorema tersebut valid untuk suatu nilai a, sehingga memenuhi $p|a^p - a$.
  Selanjutnya perlu dibuktikan bahwa $p|(a+1)^p - (a+1)$. Untuk membuktikan hal ini, perhatikan bahwa :

  $$(a+1)^p - (a+1) = a^p + \sum_{i=1}^{p-1} \binom{p}{i} a^{p-i} + 1 - (a+1)$$

  atau dapat ditulis menjadi :

  $$(a+1)^p - (a+1) = (a^p - a) + \sum_{i=1}^{p-1} \binom{p}{i} a^{p-i}$$

  Karena $p|\binom{p}{i}$ untuk $1 \le i \le p-1$ dan berdasarkan asumsi $p|a^p - a$, maka dapat disimpulkan $p \mid (a+1)^p - (a+1)$.

- **Pembuktian Menggunakan Kongruensi**
  Kita dapat mengalikan kongruensi, untuk setiap $i = 1,..,n$, dan $c_i \equiv d_i \ (mod\ p)$, sehingga

  $$c_1 \cdot c_2 \cdots c_n \equiv d_1 \cdot d_2 \cdots d_n \ (mod\ p) \qquad (1)$$

  Kemudian, misalkan $\gcd(a,p) = 1$. Kita bentuk barisan berikut :

  $$a, 2a, 3a, \ldots, (p-1)a \qquad (2)$$

  Tidak ada dua dari suku tersebut yang kongruen dengan modulo p. Karena

  $$i \cdot a \equiv k \cdot a \ (mod\ p) \rightarrow i \equiv k \ (mod\ p) \rightarrow i = k$$

  Oleh sebab itu, setiap suku dari barisan yang dibuat kongruen tepat pada satu dari bilangan berikut :

  $$1, 2, 3, .., p-1 \qquad (3)$$

  Dengan menggunakan persamaan (1), (2), dan (3) diperoleh

  $$a^{p-1} \cdot 1 \cdot 2 \cdot \cdots \cdot (p-1) \equiv 1 \cdot 2 \cdot \cdots \cdot (p-1) \ (mod\ p)$$

  Karena p dan $(p-1)!$ saling prima, maka persamaan diatas dapat ditulis menjadi :

  $$a^{p-1} \equiv 1 \ (mod\ p)$$

- **Pembuktian Menggunakan Teori Kombinatorik**
  Misalkan terdapat mutiara-mutiara dengan banyak warna sejumlah a warna. Dari mutiara ini dibentuk kalung dengan tepat menggunakan sejumlah p mutiara. Pertama, dibentuk sebuah untaian mutiara. Terdapat a^p untaian string berbeda yang dapat dibentuk. Jika kita buang untaian tersebut yang hanya terdisi dari satu warna saja, yaitu sebanyak a. Maka terdapat sisa sebanyak $a^p - a$ untaian. Kemudian, ujung dari tiap untaian tersebut untuk mendapatkan kalung. Kita dapat melihat bahwa dua untaian yang dibedakan oleh hanya sebuah permutasi siklis dari mutiaranya menghasilkan kalung yang tak dapat dibedakan. Namun. Terdapat sejumlah p permutasi siklis dari p mutiara pada sebuah untaian. Oleh sebab itu, banyak kalung yang berbeda adalah sejumlah $\frac{a^p - a}{p}$. Karena banyak kalung ini merepresentasikan bilangan bulat, maka dapat disimpulkan bahwa $p|a^p - a$.

Sebagai catatan sejarah, sekitar 500 tahun sebelum masehi, matematikawan cina sudah mengetahui bahwa jika p adalah sebuah bilangan prima, maka berlaku $2^p \equiv 2 \ (mod\ p)$, atau lebih dikenal dengan nama Hipotesis China. Hal ini merupakan kasus khusus dari Fermat's Little Theorem. Namun, mereka dan juga Leibniz ratusan tahun kemudian berpikir bahwa konvers dari teorema tersebut benar, atau dalam hal ini, jika $2^n \equiv 2 \ (mod\ n)$, maka n haruslah bilangan prima. Counterexample terkecil dari keyakinan mereka tersebut adalah $n = 341 = 11 \cdot 31$, dimana :

$$2^{341} \equiv 2^{10 \cdot 34} \cdot 2 \equiv 1^{34} \cdot 2 \equiv 2 \ (mod\ 341)$$

Bilangan yang memiliki sifat seperti 341 tersebut disebut *pseudoprime* terhadap basis 2. Beberapa bilangan juga merupakan *pseudoprimes* terhadap semu basis. Bilangan ini disebut *absolute pseudoprimes* atau *Carmichael Numbers. Carmichael Number* yang terkecil yaitu 561.

2.2 Contoh Penggunaan Fermat's Little Theorem

Berikut beberapa permasalan yang dapat diselesaikan dengan bantuan Fermat's Little Theorem :

Contoh 2.2.1 Hitung $5^{2007} \ (mod\ 41)$

Jawab:
Karena 5 dan 41 adalah bilangan prima, maka menurut Fermat's Little Theorem berlaku : $5^{40} \equiv 1 \ (mod\ 41)$
Karena $5^{2007} = 5^{40 \cdot 50} \cdot 5^7$, maka
$5^{2007} \equiv 5^{40 \cdot 50} \cdot 5^7 \equiv (5^{40})^{50} \cdot 5^7 \equiv (1)^{50} \cdot 5^7 \ (mod\ 41)$
Sehingga tinggal dihitung $5^7 \ (mod\ 41)$.
Karena $5^6 \equiv 4 \ (mod\ 41)$, maka $5^7 \equiv 20 \ (mod\ 41)$
Jadi $5^{2007} \ (mod\ 41) = 20$.

Contoh 2.2.2 Buktikan bahwa pangkat 8 dari sebarang bilangan selalu berbentuk $17m$ atau $17m \pm 1$ untuk suatu m bilangan bulat.

Jawab:

Misal N sebarang bilangan, akan dicari N^8. Jika N kelipatan 17, maka N^8 juga kelipatan 17. Sehingga N^8 mempunyai bentuk $17m$.
Sekarang, perlu dibuktikan jika N saling prima dengan 17. Karena N bilangan prima, maka menurut Fermat's Little Theorem berlaku :
$N^{16} \equiv 1 \ (mod\ 17)$
atau
$N^{16} - 1 \equiv 0 \ (mod\ 17)$
$(N^8 - 1)(N^8 + 1) \equiv 0 \ (mod\ 17)$

Jadi, $N^8 \equiv \pm 1 \ (mod\ 17)$, sehingga $N^8 = 17m \pm 1$ untuk suatu bilangan bulat m.

Contoh 2.2.3 Buktikan bahwa $n^{37} - n$ habis dibagi 1919190 untuk semua bilangan asli n.

Jawab:
Kita harus menguraikan bilangan 1919190 dalam faktor primanya, yaitu :

$$1919190 = 2 \times 3 \times 5 \times 7 \times 13 \times 19 \times 37$$

Selanjutnya, berdasarkan Fermat's Little Theorem, kita mempunyai :

$$n^{k-1} - 1 \equiv 0 \ (mod\ k)$$

Atau

$$n^k - n \equiv 0 \ (mod\ k)$$

Dengan $k = 2, 3, 5, 7, 13, 19, 37$. Perhatikan bahwa $n - 1$ membagi $n^{36} - 1$. Karena $n^{36} - 1 = (n^2)^{18} - 1$, maka $n^2 - 1$ habis membagi $n^{36} - 1$. Hal yang sama untuk $n^3 - 1$, $n^4 - 1, n^6 - 1, n^{12} - 1$, $n^{18} - 1$ semuanya habis membagi $n^{36} - 1$. Jadi $n^k - n$ habis membagi $n^{37} - n$ untuk $k = 2, 3, 5, 7, 13, 19, 37$.
Dengan demikian, menurut Chinese Remainder Theorem, maka :
$n^{37} - n \equiv 0 \ (mod\ 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 19 \cdot 37)$
$n^{37} - n \equiv 0 \ (mod\ 1919190)$

Contoh 2.2.4 Misalkan p adalah suatu bilangan prima dengan bentuk $3k + 2$ dimana membagi $a^2 + ab + b^2$ untuk suatu bilangan bulat a dan b. Buktikan bahwa a dan b keduanya habis dibagi oleh p.

Jawab :
Kita akan melakukan pendekatan secara tidak langsung, yaitu dengan mengasumsikan bahwa p tidak habis membagi a. Karena p habis membagi $a^2 + ab + b^2$, p juga habis membagi $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$. Sehingga $a^3 \equiv b^3 \ (mod\ p)$. Hal ini dapat diperumum menjadi $a^{3k} \equiv b^{3k} \ (mod\ p)$. Oleh karena itu, p juga tidak habis membagi b. Dengan menggunakan Fermat's Little Theorem, menghasilkan :
$a^{p-1} \equiv b^{p-1} \equiv 1 \ (mod\ p)$
atau
$a^{3k+1} \equiv b^{3k+1} (mod\ p)$
Karena p relatif prima terhadap a, maka dapat diputuskan bahwa
$a \equiv b \ (mod\ p)$
Hal ini, dengan mengingat sebelumnya bahwa $(a^2 + ab + b^2) \equiv 0 \ (mod\ p)$ mengakibatkan $3a^2 \equiv 0 \ (mod\ p)$. Karena $p \neq 3$, ini mengindikasikan bahwa p habis membagi a, yang menghasilkan sebuah kontradiksi dari asumsi awal.

Contoh 2.2.5 Misalkan p suatu bilangan prima. Tunjukkan bahwa terdapat tak hingga banyaknya bilangan bulat positif n sehingga p habis membagi $2^n - n$

Jawab:
Jika $p = 2$, trivial bahwa p habis membagi $2^n - n$ untuk setiap n bilangan bulat positif genap. Kita asumsikan bahwa p adalah ganjil. Dengan menggunakan Fermat's Little Theorem, didapat $2^{p-1} \equiv 1 \ (mod\ p)$ atau dapat ditulis menjadi $2^{(p-1)2k} \equiv 1 \ (mod\ p)$ Kemudian perhatikan bahwa $(p-1)^{2k} \equiv 1 \ (mod\ p)$. Sehingga $2^{(p-1)2k} \equiv 1 \equiv (p-1)^{2k} (mod\ p)$, yang secara eksplisit menunjukkan bahwa p habis membagi $2^n - n$ untuk $n = (p-1)^{2k}$.

3. Aplikasi Fermat's Little Theorem pada RSA

Algoritma RSA dijabarkan oleh tiga orang yang berasal dari MIT (Massachusetts Institute of Technology), yaitu Ron Rivest, Adi Shamir, dan Len Adleman pada tahun 1977. RSA mendasarkan proses enkripsi dan deksripsi pada konsep bilangan prima dan aritmetika modulo. Baik kunci ekripsi maupun kunci dekripsi keduanya merupakan bilangan bulat. Kunci ekripsi ini tidak dirahasiakan dan diketahui umum (dinamakan kunci publik), namun kunci untuk dekripsi bersifat rahasia. Kunci dekripsi dibangkitkan dari beberapa buah bilangan prima bersama-sama dengan kunci enkripsi. Algoritma ini dipatenkan oleh MIT pada tahun 1983 di Amerika Serikat sebagai U.S. Patent 4405829. Paten ini berlaku hingga 21 September 2000.

Langkah-langkah dari Algoritma RSA adalah sebagai berikut :

1. Ambil dua bilangan prima sembarang, misalkan a dan b. Jaga kerahasaian a dan b ini.
2. Hitung $n = a \times b$. Besaran n ini tidak dirahasiakan.
3. Hitung $m = (a-1) \times (b-1)$. Sekali m telah dihitung, a dan b dapat dihapus untuk mencegah diketahuinya oleh pihak lain.
4. Pilih sebuah bilangan bulat untuk kunci public, sebut namanya e, yang relatif prima terhadap m.
5. Bangkitkan kunci dekripsi, d, dengan kekongruenan $ed \equiv 1 \ (mod\ m)$. Lakukan enkripsi terhadap isi pesan dengan persamaan $c_i = {p_i}^e \ mod\ n$, yang dalam hal ini p_i adalah blok plainteks, c_i adalah

chiperteks yang diperoleh, dan e adalah kunci enkripsi (kunci publik). Harus dipernuhi persyaratan bahwa nilai p_i harus teletak dalam himpunan nilai $0, 1, 2, \ldots, n-1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan.

6. Proses dekripsi dilakukan dengan menggunakan persamaan $p_i = {c_i}^d \bmod n$, yang dalam hal ini d adalah kunci dekripsi.

Dua bilangan prima yang diambil, yaitu a dan b, seharusnya secara kasar memiliki ukuran yang sama, namun tidak begitu dekat. Nilai dari a dan b ini dapat dicari dengan mencoba-coba bilangan bulat yang mendekati $\sqrt{n}$. Biasanya $b < a < 2b$. Digit dari a dan b dapat dihasilkan dengan random sehingga bilangan tersebut teruji keprimaanya dengan menggunakan Fermat's Little Theorem. Tidak ada aturan yang jelas dalam memperhatikan bagaimana nilai e dipilih selain e harus relatif prima terhadap m. Namun, terdapat nilai standar untuk memilih e ini yang biasanya digunakan karena mempercepat proses perhitungan. Nilai decoding d dihitung menggunakan algoritma extended euclid.

Teorema yang merupakan dasar dari Algoritma RSA menyatakan bahwa untuk sembarang dua buah bilangan prima, misal p dan q, dan misal terdapat bilangan bulat x sehingga x saling prima terhadap p dan q. Maka berlaku :

$$x^{(p-1)(q-1)} \equiv 1 \ (mod\ pq)$$

Hal ini dapat dibuktikan sebagai akibat dari Fermat's Little Theorem. Berikut pembuktiannya :
Karena $GCD(x, p) = 1$, maka juga berlaku $GCD(x^{q-1}, p) = 1$. Dengan mengunakan Fermat's Little Theorem diperoleh
$x^{(q-1)(p-1)} \equiv 1 \ (mod\ p)$.
Dengan melakukan hal yang sama untuk q, diperoleh $x^{(q-1)(p-1)} \equiv 1 \ (mod\ q)$.
Dari kedua persamaan yang diperoleh, dapat dinyatakan bahwa terdapat bilangan bulat k dan k' sehingga memenuhi :
$kp = (x^{q-1})^{p-1} - 1$, dan
$k'q = (x^{q-1})^{p-1} - 1$
atau dalam hal ini p dan q merupakan faktor dari $(x^{q-1})^{p-1} - 1$. Sehingga, karena p dan q keduanya merupakan bilangan prima, maka $pq|(x^{q-1})^{p-1} - 1$. Atau dapat kita tulis :
$(x^{q-1})^{p-1} = 1 \ (mod\ pq)$.
Hal ini menunjukkan bahwa proses pendekripsian pada Algoritma RSA valid.

Sebagai ilustrasi, berikut gambaran dari RSA :
Misal Andi ingin mengizinkan Budi untuk mengirimkan pesan pribadi padanya. Andi melakukan langkah-langkah berikut untuk membuat pasangan kunci publik dan kunci privat :

1. Ambil dua bilangan prima sembarang, misalkan a dan b. Jaga kerahasaian a dan b ini.
2. Hitung $n = a \times b$. Besaran n ini tidak dirahasiakan.
3. Hitung $m = (a-1) \times (b-1)$. Sekali m telah dihitung, a dan b dapat dihapus untuk mencegah diketahuinya oleh pihak lain.
4. Pilih sebuah bilangan bulat untuk kunci public, sebut namanya e, yang relatif prima terhadap m.
5. Bangkitkan kunci dekripsi, d, dengan kekongruenan $ed \equiv 1 \ (mod\ m)$.

Kemudian Andi mengirimkan kunci publik pada Budi, dan tetap merahasiakan kunci privat yang digunakan. Kemudian, misalkan Budi ingin mengirim pesan A ke Andi. Budi mengubah A menjadi angka $x < n$ menggunakan protokol yang sebelumnya telah disepakati dan dikenal sebagai *padding scheme*. *Padding Scheme*
Ini harus dibangun secara hati-hati agar tidak menimbulkan masalah keamanan. Dari sini, Budi memiliki x dan mengetahui n dan e yang telah diberikan oleh Andi. Kemudian, Budi menghitung *chipertext c* yang terkait pada x, yaitu :
$c = x^e \bmod n$

Setelah itu, Andi menerima c dari Budi, dan mengetahui kunci privat yang digunakan olehnya. Andi kemudian mambangkitkan n dari c yaitu dengan perhitungan :
$x = c^d \bmod n$

Hasil ini akan memberikan nilai x, sehingga Andi dapat mengembalikan pesan semula.

Keamanan dari RSA ini bergantung pada dua asumsi. Pertama, satu-satunya cara untuk mendapatkan d dari n dan e adalah memfaktorkan n. Kedua, tidak ada algoritma yang efisien untuk memfaktorkan n ini. Sejauh yang kita tahu, kedua asumsi tersebut adalah benar, namun keduanya belum dilakukan pembuktian. Pada tahun 1993, Richard Shor mengumumkan sebuah algoritma untuk memfaktorkan bilangan bulat dengan waktu yang dibutuhkan bersifat polinomial pada komputer kuantum. Jadi, sistem RSA sampai saat ini masih aman untuk digunakan, walaupun kemajuan dalam komputasi membuatnya menjadi tidak aman di masa yang akan datang.

Pada tahun 2005, bilangan faktorisasi terbesar yang digunakan secara umum yaitu sepanjang 663 bit dengan menggunakan distribusi murakhir. Kunci RSA pada umumnya sepanjang 1024-2048 bit. Beberapa pakar meyakini bahwa kunci 1024-bit ada kemungkinan dipecahkan pada waktu dekat, namun tak seorangpun berpendapat bahwa kunci sepanjang 2048-bit akan pecah pada masa depan dengan terprediksi.

Untuk menghindari serangan pada keamanan dari RSA ini, maka sistem harus diimplemantasikan dengan hati-hati. Sebagai contoh, mengkonversikan setiap karakter dari suatu pesan ke suatu bilangan dan meng-encode bilangan tersebut akan menghasilkan kode yang mudah untuk dipecahkan. Karena bilangan dari karakter tersebut

kecil, seorang penyerang dapat dengan mudah menggunakan kunci publik untuk mengenkripsikan semua kemungkinan kode. Masalah ini dapat diselesaikan dengan mengkonversikan pesan kedalam suatu bilangan untuk dilakukan encoding atau dengan menggunakan *padding schemes* untuk mengkonversikan bilangan kecil ke bilangan yang lebih besar.

Penemu algoritma ini menyarankan nilai a dan b yang diambil (dua bilangan prima) panjangnya lebih dari 100 digit. Dengan demikian hasil $n = a \times b$ akan berukuran lebih dari 200 digit sehingga susah untuk dilakukan pemfaktoran dari n, karena menurut Rivest dan temannya, untuk mencari faktor dari n ini akan membutuhkan waktu komputasi selama 4 milyar tahun, dengan asumsi bahwa algoritma pemfaktoran yang digunakan merupakan algoritma yang tercepat saat ini dan komputer yang digunakan memiliki kecepan 1 milidetik.

IV. KESIMPULAN

Dari semua yang telah dipaparkan sebelumnya, ada beberapa kesimpulan yang dapat ditarik dari makalah ini, yaitu :

1. Dalam ilmu teori bilangan, Fermat's Little Theorem merupakan teorema dasar yang penting untuk dipahami. Teorema ini dapat digunakan untuk menguji keprimaan suatu bilangan dan sebagai dasar dari algoritma RSA. Bahkan untuk kasus ekstrim, seperti untuk membuktikan Euler's Theorem, Fermat's Little Theorem dapat digunakan. Walaupun Fermat's Little Theorem merupakan kasus khusus dari Euler's Thorem.
2. Bilangan-bilangan komposit yang lolos dari uji keprimaan dengan Fermat's Little Theorem dinamakan *fermat pseudoprimes*. Bilangan *pseudoprime* terhadap semua basis dinamakan *Carmichael Number*. Namun, bilangan-bilangan ini relatif jarang.
3. Tingkat keamanan dari algoritma RSA untuk sampai saat ini masih dapat dikatakan terjamin.

REFERENSI

[1] Engel, Arthur. *Problem-Solving Strategies*. New York : Springer-Verlag. 1998.

[2] Kisacanin, Branislav. *Mathematical Problems and Proofs – Combinatorics, Number Theory, and Geometry*. Kluwer Academic Publishers. 2002.

[3] Budhi, Wono Setya. *Langkah Awal Menuju ke Olimpiade Matematika*.Jakarta : CV. Ricardo. 2005.

[4] Titu Andreescu, Dorin Andrica, Zuming Feng. *104 Number Theory Problems – From the Training of USA IMO Team*. Birkhauser Boston. 2007.

[5] Munir, Rinaldi. *Diktat Kuliah Struktur Diskrit*. Bandung : Program Studi Teknik Informatika, Institut Teknologi Bandung. 2008.

[6] http://id.wikipedia.org/wiki/RSA. Tanggal akses : 18 Desember 2009. Pukul 17.41.

[7] http://en.wikipedia.org/wiki/Chinese_hypothesis. Tanggal akses : 18 Desember 2009. Pukul 18.11.