Pengujian Bilangan Prima

Yus Gias Vembrina / 13503042
if13042@students.if.itb.ac.id

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Abstrak

Sebagian besar bilangan yang digunakan dalam kriptografi kunci publik menggunakan bilangan prima sebagai salah satu parameternya. Bilangan prima yang disarankan adalah bilangan prima yang berukuran sangat besar, terdiri lebih dari seratus angka, bahkan ribuan angka. Karena pola kemunculan bilangan prima dalam barisan bilangan sampai dengan saat ini belum dapat dipahami manusia, dibutuhkan suatu cara untuk mengetahui sebuah bilangan termasuk bilangan prima atau tidak. Inilah yang dikenal dengan pengujian bilangan prima.

Pengujian bilangan prima adalah sebuah pengujian untuk dapat menentukan sebuah bilangan termasuk bilangan prima atau tidak. Pengujian ini terdiri dari dua jenis pengujian, yaitu deterministik dan probabilistik. Pengujian deterministik dapat menentukan secara pasti sebuah bilangan merupakan bilangan prima atau tidak. Contoh dari pengujian jenis ini adalah pembuktian keprimaan Pratt (Pratt's *primality proving*), pengujian Lucas-Lehmer (Lucas-Lehmer *test*), dan pengujian AKS (AKS *primality test*). Pengujian probabilistik, secara umum, lebih cepat daripada pengujian deterministik. Tetapi, pengujian probabilistik hanya dapat menjamin sebuah bilangan mungkin prima. Bilangan tersebut dapat disebut sebagai bilangan prima setelah diuji secara deteministik. Contoh dari pengujian probabilistik adalah pengujian Fermat (Fermat *primality test*), pengujian Solovay-Strassen (Solovay-Strassen *primality test*), dan pengujian Rabin-Miller (Rabin-Miller *primality test*).

Kata kunci: pengujian bilangan prima

1. Pendahuluan

Teori bilangan telah menjadi sebuah ilmu yang dipelajari sejak zaman Yunani kuno. Ketertarikan orang terhadap ilmu ini semakin besar lagi dalam beberapa dekade belakangan ini seiring dengan ditemukannya kriptografi kunci publik. Kriptografi kunci publik memiliki beberapa kelebihan dibandingkan dengan kriptografi kunci simetri [1] [15], atau kriptografi tradisional yang telah dikenal sejak zaman Romawi.

Salah satu kebutuhan dari sebuah sistem kriptografi adalah pesan harus dengan mudah dapat didekripsi oleh pihak yang berhak dan sulit didekripsi oleh pihak lain. Keamanan kriptografi kunci publik berlandaskan pada kerumitan memecahkan perhitungan matematis. Teori bilangan muncul sebagai sebuah sumber untuk memenuhi keperluan pengamanan tersebut. Sebagai contoh, pemfaktoran bilangan menjadi tulang punggung algoritma RSA [1] [21] dan logaritma diskrit menjadi landasan bagi algoritma ElGamal dan algoritma pertukaran kunci Diffie-Hellman [1] [34] [23]. Masalah matematis lain yang juga penting dalam pengimplementasian kriptografi kunci publik adalah pengujian bilangan prima [8].

Tulisan ini merupakan ulasan mengenai pengujian bilangan prima. Pada bagian 2 akan dibahas mengenai bilangan prima. Pada bagian 3 akan dibahas mengenai beberapa cara atau metode pengujian bilangan prima. Pada bagian 4 akan dibahas mengenai percobaan implementasi beberapa algoritma pengujian bilangan prima. Dan, pada bagian 5 disajikan tabel besar yang menggambarkan beragam algoritma pengujian bilangan prima.

2. Bilangan Prima

Bilangan prima adalah bilangan asli yang tepat hanya memiliki dua faktor, yaitu 1 dan bilangan itu sendiri.

Sampai dengan abad kesembilan belas Masehi kebanyakan matematikawan menganggap 1 sebagai bilangan prima. Pada waktu itu, sebagian besar tulisan yang dihasilkan masih memasukkan 1 sebagai bilangan prima yang sah. Perubahan yang menmbawa 1 tidak lagi dianggap sebagai bilangan prima adalah adanya kebutuhan untuk dapat menyatakan

”setiap angka dapat difaktorkan menjadi bilangan prima yang unik”.

Beberapa sifat bilangan prima

1. Semua bilangan prima berakhiran 1, 3, 7, atau 9, kecuali untuk dua bilangan prima , yaitu 2 dan 5 (bilangan berakhiran 0, 2, 4, 6, atau 8 merupakan kelipatan 2 dan bilangan berakhiran 5 merupakan kelipatan 5).

2. Jika p adalah bilangan prima dan p membagi ab yang merupakan hasil kali dua bilangan bulat, maka p membagi a atau p membagi b. Hal ini dibuktikan oleh Euclid dan dikenal sebagai teorema Euclid (Euclid’s *lemma*).

3. p adalah bilangan prima jika dan hanya jika $\varphi(p) = n - 1$. ($\varphi(p)$ adalah Euclid’s *totient*)

4. Jika p adalah bilangan prima dan a adalah sembarang bilangan bulat, maka $a^p - a$ habis dibagi oleh p. (teorema Fermat (Fermat’s *little theorem*))

5. Jika p adalah bilangan prima selain 2 dan 5, maka $1/p$ selalu menghasilkan bilangan dengan angka desimal berulang yang perulangannya terjadi dalam perioda $p-1$ atau faktor dari $p-1$.

6. p adalah bilangan prima jika dan hanya jika $(p-1)!+1$ habis dibagi oleh p. (teorema Wilson)

7. Jika n adalah bilangan bulat positif lebih besar dari 1, maka akan selalu ada bilangan prima p dengan batasan $n < p < 2n$. (postulat Bertrand)

8. Jika $p > 1$, polinom $x^{p-1} + x^{p-2} + \ldots + 1$ tidak dapat difaktorkan jika dan hanya jika p adalah bilangan prima.

9. Jika p adalah bilangan prima lebih besar dari 6, maka p mod 6 menghasilkan 1 atau 5 dan p mod 30 menghasilkan 1, 7, 11, 13, 17, 19, 23, atau 29.

Banyaknya bilangan prima

Sejak 2300 tahun yang lalu telah diketahui bahwa tidak ada bilangan prima terbesar yang diketahui. Hal ini dibuktikan oleh Euclid dengan cara kontradiksi. [4] Misalnya diketahui bahwa hanya ada tiga buah bilangan prima, yaitu p, q, dan r. Kalikan ketiga bilangan tadi dan tambahkan 1, $pqr + 1$. p tidak habis membagi $pqr + 1$ dan menyisakan 1 dari hasil pembagian tadi karena p habis membagi pqr. Hal yang sama juga terjadi pada q dan r. Disimpulkan bahwa $pqr + 1$ adalah bilangan prima atau memiliki faktor prima selain p, q, dan r. Dengan demikian, diperoleh bilangan prima lain yang belum termasuk ke dalam daftar bilangan prima yang dimiliki sebelumnya.

Faktor unik

Bilangan prima menjadi penting karena bilangan prima merupakan faktor-faktor yang membangun sebuah bilangan asli. Faktor-faktor ini unik untuk tiap bilangan asli. Hal ini juga dibuktikan oleh Euclid, “jika p adalah bilangan prima dan p membagi ab yang merupakan hasil kali dua bilangan bulat, maka p membagi a atau p membagi b”. [4] Misalkan p tidak habis membagi b. Ada $r > 0$ sebagai sisa dari hasil pembagian b oleh p, $b = cp + r$ dengan c sebuah bilangan bulat sebagai faktor pengali p. Sekarang, dengan p habis membagi ab, berarti p habis membagi $a(cp + r) = acp + ar$. Dengan begitu, p habis membagi ar dan $pk = ar$ dengan k sebuah bilangan bulat sebagai faktor pengali p. Diperoleh $\frac{p}{a} = \frac{r}{k}$. Akan tetapi, diketahui bahwa r lebih kecil daripada p (r adalah sisa pembagian oleh p). Sebuah bilangan lebih besar daripada 1 dapat habis membagi p dan a, tetapi tidak ada bilangan yang dapat membagi p kecuali 1 atau p itu sendiri. Oleh karena itu, p habis membagi a.

Pola

Berikut ini adalah beberapa buah bilangan prima.

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, …

"Apakah barisan bilangan prima di atas mengikuti suatu pola tertentu ataukah bilangan-bilangan prima tersebut muncul secara acak secara tidak menentu dalam barisan bilangan asli?"

Pertanyaan di atas telah dipertanyakan sejak dulu dan belum terjawab sampai sekarang. Tetapi, ternyata didapati banyak pola-pola kecil dalam barisan bilangan prima. Salah satunya adalah pola untuk bilangan prima yang merupakan hasil penjumlahan dua buah bilangan kuadrat.

2	1^2+1^2
3	Tidak bisa dipolakan
5	1^2+2^2
7	Tidak bisa dipolakan
11	Tidak bisa dipolakan
13	2^2+3^2
⋮	

Aturan yang mengikat pola tersebut adalah "sebuah bilangan prima dapat ditulis sebagai penjumlahan dari dua buah bilangan kuadrat jika dan hanya jika bilangan prima tersebut menyisakan 1 atau 2 ketika dibagi oleh 4".

Contoh pola lain: bilangan genap yang dapat ditulis sebagai penjumlahan dari dua buah bilangan prima.

2	Tidak bisa dipolakan
4	$2+2$
6	$3+3$
8	$3+5$
10	$3+7,\ 5+5$
12	$5+7$
⋮	

Asumsi Goldbach (Goldbach *conjecture*) yang terkenal menyatakan "setiap bilangan genap lebih besar dari 2 dapat ditulis sebagai penjumlahan dari dua buah bilangan prima". [4] Sejak tahun 1742, tidak seorang pun yang mampu membuktikannya, meskipun diketahui bahwa pernyataan itu benar adanya. Ada juga yang disebut sebagai teorema Vinogradov menyatakan bahwa "setiap bilangan ganjil lebih besar dari $10^{43.000}$ dapat ditulis sebagai penjumlahan dari tiga buah bilangan prima". [4]

Pola yang lain adalah segitiga Pascal.

0					1				
1				1		1			
2			1		2		1		
3		1		3		3		1	
4	1		4		6		4		1
5		1		5		10		5	
6			1		6		15		1
7		1		7		21		7	
8	1		8		28		56		1

Tiap baris dinomori mulai dari 0. Terdapat perbedaan antara baris-baris yang bernomor prima dengan baris-baris yang lain. Jika nomor baris, n, adalah bilangan prima, maka row ke-n hanya berisi bilangan yang merupakan kelipatan n, dengan mengabaikan angka 1 yang berada di sisi kiri dan kanan. Jika n bukan bilangan prima, maka bilangan-bilangan yang terdapat pada baris ke-n tersebut bukan merupakan kelipatan dari n.

Baris ke-n dalam segitiga Pascal, bilangan dengan urutan ke-k dari kiri (urutan dimulai dari 0) adalah

$$\binom{n}{k}=\frac{n!}{k!(n-k)!}=\frac{n(n-1)\dots(n-k+1)}{k(k-1)\dots 2.1}.$$

Misalkan n adalah bilangan prima. Pembilang mengandung faktor n dan semua faktor penyebut lebih kecil dari n sehingga tidak ada yang membagi faktor n. Oleh karena itu, n habis membagi $\binom{n}{k}$.

Misalkan n bukan merupakan bilangan prima. n dapat ditulis sebagai $n=pk$ dengan p adalah bilangan prima. Maka

$$\begin{aligned}\binom{n}{p} &= \frac{n(n-1)\dots(n-p+1)}{p(p-1)\dots 2.1}\\ &= \frac{pk(pk-1)\dots(pk-p+1)}{p(p-1)\dots 2.1}.\end{aligned}$$

p dalam penyebut dan pembilang saling meniadakan satu sama lain. Faktor terkecil dalam penyebut adalah $pk-p$, maka p tidak habis membagi $\binom{n}{p}$ dan begitu pula dengan n.

Dengan demikian, jika ada sebuah bilangan asli n dan ingin diketahui bahwa n adalah bilangan prima atau bukan, solusinya dapat

dicari dengan melihat baris ke-n dalam segitiga Pascal. Akan tetapi, cara ini tidaklah efisien. Pencarian seluruh faktor n masih merupakan cara yang lebih baik dibandingkan cara ini.

3. Pengujian Bilangan Prima

Definisi bilangan prima itu sendiri sudah memberikan arahan mengenai cara menentukan bilangan prima, yaitu mencari faktor bilangan yang ingin ditentukan keprimaannya dengan cara membagi bilangan tersebut dengan semua bilangan-bilangan yang lebih kecil daripada bilangan yang diuji. Cara ini telah dikenal sejak zaman Yunani kuno dan merupakan spesialisasi dari *sieve of Eratosthenes* (tahun 240 sebelum masehi) yang digunakan untuk mencari bilangan prima yang lebih kecil dari sebuah bilangan tertentu, misalkan n. Pengujian dengan cara ini tidaklah efisien. Dibutuhkan langkah sebanyak $\sqrt{n}$ untuk dapat menentukan sebuah bilangan n merupakan bilangan prima atau bukan.

Pengujian bilangan prima yang efisien seharusnya hanya membutuhkan langkah dalam jumlah yang polinomial, yaitu dalam ukuran $\lceil \log n \rceil$. Pengujian bilangan prima yang baik seharusnya memenuhi hal-hal berikut. [8]

1. Tepat, algoritma harus selalu memberikan jawaban yang tepat.
2. Umum, algoritma harus dapat memproses semua bilangan, tidak hanya bilangan dengan bentuk tertentu.
3. Cepat, algoritma harus menghabiskan waktu dalam pola polinomial.

Sebuah pengujian bilangan prima yang hampir memberikan pengujian yang efisien adalah pengujian Fermat's *little theorem*. Teorema tersebut menyatakan "jika p adalah bilangan prima dan a adalah sembarang bilangan bulat, maka $a^p - a$ habis dibagi oleh p". Cara ini secara efisien akan memeriksa keprimaan sebuah bilangan. Akan tetapi, pengujian ini tidak selalu mengeluarkan hasil yang benar, ada bilangan tidak prima yang memenuhi teorema tadi, yang disebut dengan bilangan Carmichael. [40] Meskipun demikian, Fermat's *little theorem* adalah basis dari pengujian bilangan prima lainnya.

Sejak awal teori kompleksitas di awal tahun 1960-an, ketika notasi kompleksitas diformalkan dan berbagai kelas kompleksitas ditetapkan, masalah pengujian bilangan prima diselidiki secara seksama. Disimpulkan bahwa pengujian bilangan prima termasuk ke dalam kelas co-NP. Pada tahun 1975, Pratt meneliti masalah pengujian bilangan prima dan menyimpulkan bahwa masalah ini juga termasuk ke dalam kelas NP. [15] Dengan demikian, kompleksitas masalah pengujian bilangan prima termasuk ke dalam kelas $\text{co-NP} \cap \text{NP}$.

Pada tahun 1976, Miller menghasilkan algoritma yang diturunkan dari Fermat's *little theorem* untuk melakukan pengujian bilangan prima dalam waktu polinomial dengan mengandalkan asumsi *Extended Riemann Hypothesis* (ERH). [34] ERH dipercaya benar, namun belum dapat dibuktikan kebenarannya secara formal. Tidak lama kemudian, pengujian tersebut dimodifikasi oleh Rabin untuk menghasilkan algoritma tanpa menggunakan asumsi ERH tetapi memerlukan waktu pengujian acak namun tetap polinomial. [28] Di lain pihak, Solovay dan Strassen mendapatkan algoritma yang berbeda namun tetap memerlukan waktu polinomial yang acak dengan memanfaatkan sifat bilangan prima "untuk sebuah bilangan prima p, $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \pmod p$ untuk setiap a ($\left(\frac{a}{p}\right)$ adalah simbol Jacobi)". [34] Algoritma ini juga dapat diterapkan menggunakan ERH. Sejak saat itu, sejumlah pengujian bilangan prima yang memerlukan waktu polinomial acak yang telah dicoba dibuat dengan berdasarkan pada sifat bilangan prima yang berbeda-beda.

Di saat algoritma-algoritma pengujian bilangan prima yang lain memerlukan waktu yang eksponensial, pada tahun 1983, Adlemann, Pomerance, dan Rumely berhasil membuat algoritma pengujian bilangan prima yang memerlukan waktu polinomial, yaitu $(\log n)^{(\log \log \log n)}$. [26] Algoritma yang mereka kembangkan merupakan generalisasi dari algoritma Miller dan menggunakan sifat yang lebih tegas lagi.

Pada tahun 1986, Goldwasser dan Kilian mengajukan sebuah algoritma pengujian bilangan prima dengan menggunakan kurva eliptik (*elliptic curve*) yang diharapkan memerlukan waktu polinomial untuk hampir semua masukan yang diberikan (semua masukan yang berada dalam hipotesis yang dipercaya). [21] Berdasarkan pada algoritma mereka, algoritma serupa dikembangkan oleh Atkin. [22] Adleman dan Huang memodifikasi

algoritma Goldwasser-Kilian sehingga dapat menerima semua masukan. [15]

Pada bulan Agustus 2002, Manindra Agrawal, Neeraj Kayal, dan Nitin Saxena mengajukan sebuah pengujian bilangan prima yang cepat, hanya memerlukan waktu $\log^{\frac{15}{2}} n$, dan bekerja tanpa menggunakan asumsi. [7] Tidak hanya pengujian ini tidak pernah gagal, pengujian ini juga lebih sederhana dibandingkan pengujian-pengujian bilangan prima lain yang mendekati waktu polinomial. Pengujian ini didasari sifat bilangan prima

$$(X + a)^n \bmod n = (X^n + a) \bmod n .$$

Tetapi, di sisi lain, pengujian ini juga termasuk lambat. [4] Jumlah langkah yang dilakukan dalam pengujian bilangan prima menggunakan algoritma ini bertambah sejumlah angka bilangan yang diuji dipangkatkan 12. Beberapa bulan kemudian, Lenstra memperbaiki hal ini menjadi langkah yang dilakukan tumbuh sebanyak angka bilangan yang diuji dipangkatkan 6. [5]

Lebih dalam dengan beberapa metode pengujian bilangan prima

Pengujian bilangan prima dapat dikelompokkan ke dalam dua jenis.

1. Deterministik, jika dapat menentukan secara pasti sebuah bilangan merupakan bilangan prima atau tidak. Kepastian ini diperoleh dari pembuktian matematis secara formal sehingga dapat dijamin kondisinya terpenuhi jika dan hanya jika bilangan tersebut merupakan bilangan prima.

2. Probabilistik, jika hanya dapat menjamin sebuah bilangan mungkin prima. Pengujian ini hanya menjamin bahwa bilangan mungkin prima karena tidak mencoba seluruh angka yang dapat dicobakan ke dalam persamaan, hanya mencoba beberapa angka yang dipilih secara acak.

Pengujian bilangan prima deterministik

Beberapa pengujian yang termasuk ke dalam jenis ini di antaranya adalah pembuktian keprimaan Pratt, pengujian Lucas-Lehmer, dan pengujian AKS.

Pembuktian keprimaan Pratt [39]

Teorema

Sebuah bilangan asli $m > 2$ merupakan bilangan prima jika dan hanya jika terdapat bilangan bulat a sedemikian sehingga

$$a^{m-1} = 1 \pmod m$$

dan

$$a^x \neq 1 \pmod m \quad \forall \quad x = 1,2,\ldots,m-2 .$$

Bilangan a dikenal dengan akar primitif dari bilangan prima m.

Pembuktian formal

Secara umum setelah memverifikasi persamaan di atas, tidak perlu dilakukan perhitungan sebanyak $m-2$ kali untuk $a^x \bmod m$ demi memverifikasi pertidaksaman di atas. Hanya perlu dilakukan verifikasi $a^{(m-1)/p} \neq 1 \pmod m$ untuk semua hasil pembagian $m-1$ oleh semua faktor prima dari p.

Untuk melakukan pembuktian formal, digunakan notasi

m

yang dibaca sebagai "m adalah bilangan prima" dan

(m, a, x)

yang dibaca sebagai "setiap faktor prima p dari x memenuhi $a^{(m-1)/p} \neq 1 \pmod m$".

Pembuktian ini memiliki satu axioma

$(m, a, 1)$

untuk semua bilangan bulat positif m dan a.

dan dua aturan inferensi

$(m, a, x), p \vdash (m, a, xp)$

selama $a^{(m-1)/p} \neq 1 \pmod m$ terpenuhi

dan

$(m, a, m-1) \vdash m$

selama $a^{m-1} = 1 \pmod m$ terpenuhi.

Contoh

Misalkan bilangan yang ingin diuji keprimaannya adalah 1783.

(S1)	(2,1,1)	axioma
(S2)	2	(S1)
(S3)	(3,2,1)	axioma
(S4)	(3,2,2)	(S3) dan (S2)
(S5)	3	(S4)
(S6)	(5,2,1)	axioma
(S7)	(5,2,2)	(S6) dan (S2)
(S8)	(5,2,4)	(S7) dan (S2)
(S9)	5	(S8)
(S10)	(11,2,1)	axioma
(S11)	(11,2,2)	(S10) dan (S2)
(S12)	(11,2,110)	(S11) dan (S9)
(S13)	11	(S12)
(S14)	(1783,10,1)	axioma
(S15)	(1783,10,2)	(S14) dan (S2)
(S16)	(1783,10,6)	(S15) dan (S5)
(S17)	(1783,10,18)	(S16) dan (S5)
(S18)	(1783,10,54)	(S17) dan (S5)
(S19)	(1783,10,162)	(S18) dan (S5)
(S20)	(1783,10,1782)	(S19) dan (S13)
(S21)	1783	(S20)

Pengujian Lucas-Lehmer

Teorema

Jika ada a yang lebih kecil dari n dan lebih besar dari 1 sehingga

$$a^{n-1} = 1 \pmod{n}$$

dan

$$a^{(n-1)/q} \neq 1 \pmod{n}$$

untuk semua faktor prima q dari $n-1$, maka n adalah bilangan prima. Jika tidak ada a yang memenuhi kondisi di atas, maka n bukan merupakan bilangan prima.

Pembuktian

Kebenaran dari pengujian ini mirip seperti pada pembuktian keprimaan Pratt. Secara sederhana hal ini dapat dinyatakan sebagai berikut.

Jika a memenuhi persamaan di atas, maka dapat disimpulkan bahwa a dan n relatif prima. Dan, jika a memenuhi pertidaksamaan di atas, pemangkat a yang berada di dalam Z/nZ setara dengan $n-1$. Hal tersebut menyatakan bahwa n adalah bilangan prima. Sebaliknya, jika n adalah bilangan prima, maka akan ada akar primitif dalam modulo n yang akan memenuhi persamaan dan pertidaksamaan di atas.

Pengujian AKS

Pengujian AKS ini terdiri dari langkah-langkah sebagai berikut.

masukannya adalah bilangan bulat $n > 1$

1. jika ($n = a^b$ untuk $a \in \mathrm{N}$ dan $b > 1$), n bukan bilangan prima
2. cari r terkecil sedemikian sehingga $(ak = 1 \pmod{r}) > \log^2 n$ dengan k adalah bilangan bulat yang kecil
3. jika $1 < (a,n) < n$ untuk beberapa $a \leq r$, n bukan bilangan prima
4. jika $n \leq r$, n adalah bilangan prima
5. untuk $a = 1$ sampai $\lfloor \sqrt{\phi(r)} \log n \rfloor$ lakukan
   jika
   (
   $(X\_a)^n \neq X^n + a \pmod{X^r - 1, n}$
   ), n bukan bilangan prima
6. n adalah bilangan prima

Dasar dari metode pengujian AKS ini adalah perhitungan pada segitiga Pascal yang telah diuraikan sebelumnya. Dari perhitungan tersebut dapat diturunkan bahwa untuk setiap a yang relatif prima terhadap n,

$$(X + a)^n \bmod n = (X^n + a) \bmod n$$

untuk semua nilai X, jika dan hanya jika n merupakan bilangan prima.

Akan tetapi, seperti pada segitiga Pascal, perhitungan persamaan di atas akan menjadi tidak efisien dengan membesarnya bilangan n.

Pengujian AKS berhasil mengatasi hal ini dengan melakukan perhitungan dalam modulo polinomial $X^r - 1$. Oleh karena itu, diperoleh persamaan baru

$$(X + a)^n = (X^n + a) \pmod{X^r - 1, n}$$

yang akan selalu terpenuhi jika n merupakan bilangan prima. Lebih jauh lagi, dalam pengujian ini ditunjukkan bahwa "jika n bukan merupakan bilangan prima dan dipilih

nilai yang *tepat* untuk r, maka hanya perlu dicoba untuk beberapa a sampai didapatkan

$$(X\_a)^n \neq X^n + a \pmod{X^r - 1, n}".$$

Ketika diperoleh nilai a tersebut, maka terbukti bahwa n bukan merupakan bilangan prima. Nilai a tidak dipilih secara acak. Pengujian AKS merupakan metode pengujian bilangan prima yang deterministik.

Pembuktian secara formal dari pengujian ini cukup kompleks. Lebih jelasnya dapat dilihat di [7].

Pengujian bilangan prima probabilistik

Beberapa pengujian yang termasuk ke dalam jenis ini di antaranya adalah pengujian Fermat, pengujian Solovay-Strassen, dan pengujian Rabin-Miller.

Pengujian Fermat

Teorema

Jika p adalah bilangan prima dan $1 < a < p$, maka

$$a^{p-1} = 1 \pmod{p}.$$

Cara pengujian

Pengujian bilangan prima menggunakan metode ini dilakukan dengan memilih a secara acak kemudian menguji persamaan di atas. Apabila persamaan tersebut berhasil dipenuhi maka dikatakan p adalah pseudoprima. Jika tidak, dikatakan p bukan merupakan bilangan prima.

Kekurangan

Pengujian ini hanya menghasilkan kemungkinan pseudoprima karena tidak semua a diuji, hanya dipilih secara acak, dan ada bilangan yang memenuhi persamaan di atas padahal bilangan tersebut bukanlah merupakan bilangan prima, meskipun semua nilai a telah dicoba dievaluasi. Bilangan tersebut dikenal dengan bilangan Charmichael. [41] Inilah kekurangan yang dimiliki oleh metode pengujian ini.

Pengujian Solovay-Strassen

Teorema

Euler membuktikan bahwa untuk bilangan prima p dan sembarang bilangan bulat a, dipenuhi kondisi

$$a^{(p-1)/2} = \left(\frac{a}{p}\right) \pmod{p}$$

dengan $\left(\frac{a}{p}\right)$ adalah simbol Legendre.

Simbol Jacobi merupakan generalisasi dari simbol Legendre dari $\left(\frac{a}{n}\right)$ dengan n adalah sembarang bilangan bulat ganjil.

Cara pengujian

Pengujian yang dilakukan persis seperti pengujian yang dilakukan dalam pengujian Fermat, dipilih a secara acak dan kemudian nilai tersebut dievaluasi ke dalam persamaan. Apabila persamaan tersebut berhasil dipenuhi maka dikatakan p adalah pseudoprima. Jika tidak, dikatakan p bukan merupakan bilangan prima.

Kekurangan

Metode ini memiliki kekurangan yang sama dengan pengujian Fermat. Akan tetapi, metode ini berhasil menekan tingkat kesalahan pengklasifikasian menjadi ½ kali kesalahan pada pengujian Fermat.

Ada pun probabilitas pengujian Solovay-Strassen ini salah mengklasifikasikan bilangan adalah 2^{-k} dengan k adalah jumlah putaran dilakukannya pengujian bilangan prima dengan nilai a yang berbeda-beda. Pengujian sebanyak 100 kali dirasakan cukup untuk memperkecil tingkat kesalahan yang mungkin terjadi, meskipun tidak bisa menjadi jaminan 100%.

Pengujian Rabin-Miller

Teorema

Diawali dengan *lemma* mengenai akar kuadrat dalam daerah terbatas Z_p, dengan p adalah bilangan prima. Sudah tentu pengakarkuadratan modulo p akan menghasilkan 1 atau -1. Ini dapat diilustrasikan sebagai berikut.

$$x^2 = 1 \pmod{p}$$

$$(x-1)(x+1) = 0 \pmod{p}$$

Misalkan n adalah bilangan prima ganjil, maka $n-1$ dapat ditulis sebagai $2^s.d$ dengan s adalah bilangan bulat dan d adalah bilangan ganjil. Maka salah satu dari persamaan di bawah harus dipenuhi oleh semua $a \in (Z/nZ)$.

$$a^d = 1 \pmod{n}$$

$$a^{2^r.d} = -1 \pmod{n} \quad \exists \quad 0 \le r \le s-1$$

Cara pengujian

Sama seperti pada dua metode pengujian sebelumnya, dipilih a secara acak dan kemudian nilai tersebut dievaluasi ke dalam persamaan. Apabila salah satu persamaan di atas berhasil dipenuhi maka dikatakan p adalah pseudoprima. Jika tidak, dikatakan p bukan merupakan bilangan prima.

Kekurangan

Metode berhasil menekan tingkat kesalahan pengklasifikasian lebih jauh lagi, yaitu menjadi $\frac{1}{4}$ kali kesalahan pada pengujian Fermat.

Sedangkan tingkat kesalahan yang mungkin terjadi adalah 4^{-k} dengan k adalah jumlah putaran dilakukannya pengujian bilangan prima dengan nilai a yang berbeda-beda. Seperti pada pengujian Solovay-Strassen, pengujian yang dilakukan berulang-ulang dapat memperkecil kemungkinan terjadinya kesalahan pengklasifikasian bilangan.

4. Percobaan

Percobaan dilakukan menggunakan kelas bilangan bulat besar yang dibuat sendiri. Karena ketebatasan-keterbatasan yang dimiliki oleh kelas ini (belum dapat menghitung logaritma diskrit, belum dapat melakukan pemfaktoran suatu bilangan, proses perhitungan yang mungkin belum optimal), percobaan hanya dilakukan terhadap pengujian bilangan prima probabilistik yang dibahas lebih lanjut dalam bagian 3 yang operasi perhitungan telah dapat dilakukan oleh kelas tersebut.

Percobaan ini dilakukan untuk melihat kebenaran hasil dari metoda dalam melakukan pengujian bilangan prima. Hasil perolehannya dibandingkan dengan hasil pengujian bilangan prima dengan menggunakan metode yang naif, mencoba mencari faktor dari semua kemungkinan yang ada.

Hasil percobaan menunjukkan hasil yang sama seperti yang terdapat dalam uraian di bagian 3. Metode-metode probabilistik yang diuji dapat saja salah menyatakan keprimaan sebuah bilangan. Akan tetapi, hal ini jarang terjadi. Dengan memperbanyak jumlah putaran pengujian dalam masing-masing metode dapat menghindarkan dari kesalahan pengklasifikasian ini.

5. Rangkuman

Pembahasan mengenai pegujian bilangan prima yang terdapat dalam bagian 3 hanya mencakup sebagian kecil dari banyak algoritma pengujian bilangan prima yang telah diajukan. Bernstein merangkum sejumlah algoritma bilangan prima dengan cukup lengkap disertai dengan perbandingan kompleksitas dari algoritma-algoritma yang ada. [2] Tabel perbandingan dapat dilihat pada Tabel 1. Adapun keterangan untuk masing-masing kolom dalam tabel tersebut adalah sebagai berikut.

1. Metode, rangkuman dari teorema yang digunakan oleh metode yang bersangkutan.
2. Efek pembuktian, informasi yang diberikan oleh metode mengenai bilangan masukan.
3. Pembuktian untuk, bilangan masukan yang dapat diterima oleh metode.
4. Waktu klarifikasi pembuktian, kompleksitas dalam memerika pembuktian keprimaan bilangan masukan.
5. Waktu mencari pembuktian, kompleksitas dalam mencari parameter untuk pembuktian keprimaan bilangan masukan.

Tabel 1 Metode-metode pengujian bilangan prima

Metode	Efek pembuktian	Pembuktian untuk	Waktu klarifikasi pembuktian	Waktu mencari pembuktian
pembuktian ketidakprimaan menggunakan faktorisasi: jika b membagi n dan $1 < b < n$ maka n tidak prima	membuktikan ketidakprimaan	setiap bilangan bukan prima	$(\log n)^{1+o(1)}$	sangat lambat; tetapi $(\log n)^{o(1)}$ untuk kebanyakan n
pembuktian ketidakprimaan menggunakan Fermat's *little theorem*: jika n tidak membagi $b^n - b$ maka n tidak prima	membuktikan ketidakprimaan	hampir setiap bilangan bukan prima; meskipun demikian, terdapat tak hingga bilangan bukan prima yang tidak memenuhi metode ([17])	$(\log n)^{2+o(1)}$	acak $(\log n)^{2+o(1)}$
jika n tidak membagi kebanyakan faktor dari $b^n - b$, maka n tidak prima ([40])	membuktikan ketidakprimaan	setiap bilangan bukan prima	$(\log n)^{2+o(1)}$	acak $(\log n)^{2+o(1)}$ ([37], secara mandiri [29], secara mandiri [27]; varian lain yang lebih rendah kualitasnya [38], [34]; varian lain [12], [8], [11], [9], [6])
asumsi pengujian keprimaan: jika n adalah *b-sprp* untuk setiap bilangan prima b di antara 1 dan $\lceil \log n \rceil^2$, maka n mungkin bilangan prima ([35])	asumsi pengujian keprimaan; asumsi mengikuti GRH ([24]);	setiap bilangan prima	$(\log n)^{4+o(1)}$	instan
jika n adalah *b-sprp*, untuk bilangan prima ke $2\lceil \log n \rceil$, maka n mungkin prima ([17])	asumsi membuktikan ketidak pertian	setiap bilangan prima	$(\log n)^{3+o(1)}$	instan

Metode	Efek pembuktian	Pembuktian untuk	Waktu klarifikasi pembuktian	Waktu mencari pembuktian
jika n adalah *2-sprp* dan telah melewati pengujian *quadratic*, maka n mungkin prima ([30], [31]; varian lain yang juga mencakup *cubic test*: [13])	asumsi pengujian keprimaan; asumsi tidak beralasan untuk n yang sangat besar ([25]) namun tidak ada penyangkal yang ditemukan	setiap bilangan prima	$(\log n)^{2+o(1)}$	instan
membuktikan keprimaan dengan menggunakan faktor *unit-group*: jika $b^{n-1} = 1$ dalam Z/n, dan $b^{(n-1)/q} - 1 \neq 0$ dalam Z/n untuk setiap q, maka n adalah prima	membuktikan keprimaan	setiap bilangan prima	paling lama $(\log n)^{3+o(1)}$; seringkali $(\log n)^{2+o(1)}$	sangat lambat; diasumsikan sebagai $(\log n)^{o(1)}$ untuk n yang tak hingga banyaknya
jika $b^{n-1} = 1$ dalam Z/n, F adalah faktor dari $n-1$ dan $b^{(n-1)/q} - 1$ adalah unit dalam Z/n untuk setiap q yang merupakan faktor dari F, maka setiap faktor n berada dalam bentuk $\{1, F+1, \ldots\}$, jadi jika $(F+1)^2 > n$ maka n adalah bilangan prima ([33])	membuktikan keprimaan	setiap bilangan prima	paling lama $(\log n)^{3+o(1)}$; seringkali $(\log n)^{2+o(1)}$	sangat lambat; $(\log n)^{o(1)}$ untuk masukan n yang tak terhingga ([20], [19], [14])
pengujian Pocklington menggunakan ekstensi *quadratic* dari Z/n	membuktikan keprimaan	setiap bilangan prima	paling lama $(\log n)^{3+o(1)}$; seringkali $(\log n)^{2+o(1)}$	sangat lambat
pengujian Pocklington menggunakan derajat yang lebih tinggi dari Z/n	membuktikan keprimaan	setiap bilangan prima	$(\log n)^{(\log\log\log n)}$, menggunakan distribusi dari faktor $n^b - 1$	instan

Metode	Efek pembuktian	Pembuktian untuk	Waktu klarifikasi pembuktian	Waktu mencari pembuktian
pembuktian keprimaan menggunakan kurva eliptik: pengujian serupa yang menggunakan kurva eliptik ([21])	membuktikan keprimaan, menggunakan kurva eliptik sebagai ukuran pembatas	hampir setiap bilangan prima; diasumsikan dapat menerima setiap bilangan prima	$(\log n)^{3+o(1)}$	$(\log n)^{o(1)}$ dengan menggunakan kurva eliptik
pengujian serupa yang menggunakan kurva eliptik dengan orde yang dapat dibagi oleh bilangan besar yang merupakan perpangkatan 2	membuktikan keprimaan, menggunakan kurva eliptik sebagai ukuran pembatas	setiap bilangan prima	$(\log n)^{2+o(1)}$	sangat lambat
pengujian serupa dengan menggunakan simbol Jacobi bergenus-2 pada kurva hipereliptik	membuktikan keprimaan, menggunakan simbol Jacobi sebagai ukuran pembatas	setiap bilangan prima	selambat-lambatnya $(\log n)^{3+o(1)}$	acak $(\log n)^{o(1)}$, menggunakan distribusi bilangan prima dengan interval lebar $x^{3/4}$ di sekitar x
pengujian serupa yang menggunakan kurva eliptik dengan diskriminan kecil dan perkalian kompleks	membuktikan keprimaan, menggunakan kurva eliptik sebagai ukuran pembatas	diasumsikan dapat menerima setiap bilangan prima	selambat-lambatnya $(\log n)^{3+o(1)}$	selambat-lambatnya $(\log n)^{5+o(1)}$
pengujian serupa yang menggunakan kurva eliptik dengan diskriminan kecil, perkalian kompleks, dan penggabungan perhitungan akar kuadrat untuk banyak diskriminan	membuktikan keprimaan, menggunakan kurva eliptik sebagai ukuran pembatas	diasumsikan dapat menerima setiap bilangan prima	selambat-lambatnya $(\log n)^{3+o(1)}$	selambat-lambatnya $(\log n)^{4+o(1)}$

Metode	Efek pembuktian	Pembuktian untuk	Waktu klarifikasi pembuktian	Waktu mencari pembuktian
membuktikan keprimaan menggunakan kombinatorik: jika bisa dituliskan banyak elemen dari sebuah subgrup ekstensi bilangan prima *cyclotomic* dalam Z/n, maka n adalah perpangkatan dari bilangan prima.	membuktikan keprimaan	setiap bilangan prima	$(\log n)^{o(1)}$, menggunakan analisis fakta bahwa untuk setiap $c > 1/2$, banyak bilangan prima r yang mempunyai faktor pembagi $r-1$ di atas r^c; selambat-lambatnya $(\log n)^{12+o(1)}$, menggunakan analisis fakta bahwa banyak bilangan prima r yang mempunyai faktor pembagi $r-1$ di atas $r^{2/3}$; diasumsikan memerlukan waktu $(\log n)^{6+o(1)}$	instan
varian yang menggunakan ekstensi *arbitrary cyclotomic*	membuktikan keprimaan	setiap bilangan prima	selambat-lambatnya $(\log n)^{12+o(1)}$, menggunakan pembatasan murni dalam pendistribusian bilangan prima; selambat-lambatnya $(\log n)^{8+o(1)}$, menggunakan analisis fakta seperti di atas; diasumsikan memerlukan waktu $(\log n)^{6+o(1)}$	instan
varian yang menggunakan ekstensi *arbitrary cyclotomic* yang menggunakan batas yang lebih baik dalam penstrukturan grup	membuktikan keprimaan	setiap bilangan prima	selambat-lambatnya $(\log n)^{10,5+o(1)}$, menggunakan pembatasan murni dalam pendistribusian bilangan prima; selambat-lambatnya $(\log n)^{7,5+o(1)}$, menggunakan analisis fakta seperti di atas; diasumsikan memerlukan waktu $(\log n)^{6+o(1)}$	instan
Metode	Efek pembuktian	Pembuktian	Waktu klarifikasi	Waktu mencari

		untuk	pembuktian	pembuktian
varian yang menggunakan ekstensi Kummer secara acak	membuktikan keprimaan	setiap bilangan prima	$(\log n)^{4} + o(1)$, menggunakan distribusi dari faktor $n^{b} - 1$	acak $(\log n)^{2} + o(1)$
varian yang menggunakan periode Gauss	membuktikan keprimaan	setiap bilangan prima	$(\log n)^{6} + o(1)$, menggunakan beragam analisis fakta seperti di atas	instan
jika n gagal dalam pengujian jenis Fermat apapun dalam metode-metode ini, maka n bukanlah bilangan prima	membuktikan ketidakprimaan	setiap bilangan bukan prima	selambat-lambatnya $(\log n)^{4} + o(1)$, menggunakan beragam analisis fakta seperti di atas	selambat-lambatnya $(\log n)^{6} + o(1)$, menggunakan beragam analisis fakta seperti di atas

6. Kesimpulan

Dengan ditemukannya algoritma AKS pengujian bilangan prima yang bersifat deterministik dapat berjalan dalam waktu polinomial. Hal ini tentu saja makin mempercepat pengklarifikasian keprimaan sebuah bilangan secara pasti.

Meskipun demikian, pengujian bilangan prima, umumnya pengetahuan bilangan prima, masih terus dikaji lebih lanjut. Masih ada sifat-sifat bilangan prima yang dapat dikembangkan untuk dijadikan acuan pengujian bilangan prima. Tujuan akhir yang ingin dicapai, tentu saja, menguak pola kemunculan bilangan prima dalam barisan bilangan asli.

7. Daftar Pustaka

[1] Munir, Rinaldi. *Diktat Kuliah IF5054 Kriptografi*. 2006. Bandung: Institut Teknologi Bandung.

[2] Bernstein, Daniel J. *Distinguishing Prime Numbers from Composite Numbers: the State of the Art in 2004*. 2004.

[3] Crandall, R. dan J. Papadopoulos. *On the implementation of AKS-class primality tests*. 2003.

[4] Aaronson, Scott. *The Prime Facts: From Euclid to AKS*. 2003.

[5] Lenstra, HW. Jr. *Primality testing with cyclotomic rings*. 2002.

[6] Damgard, Ivan B. dan Gudmund Skovbjerg Frandsen. *An extended quadratic Frobenius primality test with average and worst case error estimates*. 2003.

[7] Agrawal, Manindra, Neeraj Kayal, dan Nitin Saxena. *PRIMES is in P*. 2002. Kanpur: Department of Computer Science & Engineering Indian Institute of Technology Kanpur.

[8] Grantham, Jon. *Frobenius pseudoprimes*. 2001.

[9] Müller, Siguna. *A probable prime test with very high confidence for n ≡ 1 mod 4*. 2001.

[10] Garefalakis, Theodoulos. *Primality Testing, Integer Factorization, and Discrete Logarithms*. 2000. Toronto: Department of Computer Science, University of Toronto.

[11] Müller, Siguna. *On probable prime testing and the computation of square roots mod n*. 2000.

[12] Grantham, Jon. *A probable prime test with high confidence*. 1998.

[13] Atkin, AOL. *Intelligent primality test offer*. 1998.

[14] Konyagin, S. dan C. Pomerance. *On primes recognizable in deterministic polynomial time*. 1997.

[15] Lukes, Richard F., CD. Patterson, Hugh C. Williams. *Numerical seiving defice: their history and some applicatoon*. 1995.

[16] Odlyzko, AM. *Public key cryptography*. 1994.

[17] Alford, WR., Andrew Granville, dan Carl Pomerance. *There are infinitely many Carmichael numbers*. 1994.

[18] Adleman, LM. dan MD. Huang. *Primality testing and two dimensional Abelian varieties over finite fields*. 1992.

[19] Fellows, MR. dan N. Koblitz. *Self-witnessing polynomial-time complexity and prime factorization*. 1992.
[20] Pintz, J, WL. Steiger, dan Endre S. *Infinite sets of primes with fast primality tests and quick generation of large primes*. 1989.
[21] Goldwasser, S. dan J. Kilian. *Almost all prime can be quickly certified*. 1986.
[22] Atkin, AOL. Lecture notes of a conference, boulder (colorado). 1986.
[23] ElGamal, T. *A public key cryptosystem and a signature scheme based on discrete logarithms*. 1985.
[24] Bach, Eric. *Analytic methods in the analysis and design of number-theoretic algorithms*. 1985.
[25] Pomerance, C. *Are there counter-examples to the Baillie-PSW primality test?*. 1984.
[26] Adleman, LM., C. Pomerance, dan RS. Rumely. *On distinguishing prime numbers from composite numbers*. 1983.
[27] Atkin, AOL. dan Richard G. Larson. *On a primality test of Solovay and Strassen*. 1982.
[28] Rabin, MO. *Probabilistic algorithm for testing primality*. 1980.
[29] Monier, Louis. *Evaluation and comparison of two efficient probabilistic primality testing algorithms*. 1980.
[30] Baillie, Robert dan Samuel S. Wagstaff, Jr. *Lucas pseudoprimes*. 1980.
[31] Pomerance, C., John L. Selfridge, dan Samuel S. Wagstaff, Jr. *The pseudoprimes to 25.10^9*. 1980.
[32] Rivest, RL., A. Shamir, dan LM. Adleman. *A method for obtaining digital signatures and public-key cryptosystems*. 1978.
[33] Pocklington, Henry C. *The determination of the prime or composite nature of large numbers by Fermat's theorem*. 1978.
[34] Solovay, R dan V. Strassen. *A fast Monte-Carlo test for primality*. 1977.
[35] Miller, GL. *Riemann's hypothesis and tests for primality*. 1976.
[36] Diffie, W. dan M. Hellman. *New directions in cryptography*. 1976.
[37] Rabin, MO. *Probabillistic algorithm*. 1976.
[38] Lehmer, DH. *Strong Carmichael numbers*. 1976.
[39] Pratt, VR. *Every prime has a succinct certificate*. 1975.
[40] Artjuhov, MM. *Certain criteria for primality of numbers connected with the little Fermat theorem*. 1966.
[41] Carmichael, RD. *Note on a number theory function*. 1910.