Bilangan Prima dan Aplikasinya dalam Bidang Informatika

Hanugrha Abidianto [1)] **NIM: 135 07 008**
1) Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
Email: that.kid.is.sherlock@gmail.com

Abstract *– Makalah ini membahas pengenalan bilangan prima, sejarah singkat, pencarian bilangan prima, dan aplikasi bilangan prima dalam bidang informatika khususnya dalam kriptografi, yaitu: RSA. Pokok bahasan dalam makalah ini meliputi definisi bilangan prima, sejarah singkat perkembangan bilangan prima, pencarian dan penentuan bilangan prima, dan aplikasi bilangan prima pada kriptografi RSA.*

Bilangan prima merupakan salah satu bagian dari teori bilangan yang telah menjadi objek penelitian selama berabad-abad. Salah satu hal yang membuatnya menarik adalah pola distribusi bilangan prima yang seolah acak. Oleh sebab itu, bilangan prima banyak digunakan di bidang informatika dalam proses enkripsi pesan dengan tingkat keamanan yang relatif tinggi.

Kata Kunci*: bilangan prima, distribusi, pencarian, RSA.*

1. PENDAHULUAN

Bilangan prima adalah salah satu bilangan yang misterius dalam sejarah perkembangan matematika. Pola persebaran yang acak menjadi salah satu faktor penyebab kemisteriusan bilangan ini. Banyak matematikawan mencoba untuk memecahkan persoalan distribusi bilangan ini, namun belum ada satu cara pun yang efektif dan sempurna dalam menentukan pola distribusi bilangan prima. Oleh karena itu, pencarian atas bilangan ini masih sulit dilakukan, khususnya pada rentang bilangan yang cukup besar. Namun, pola distribusi ini memberikan keuntungan bagi beberapa pihak, salah satunya pada bidang kriptografi. Apabila bilangan prima dijadikan kunci pada proses enkripsi pesan, *brute force* untuk memecahkan pesan tersebut tidaklah mudah sehingga cara ini mempunyai tingkat keamanan yang relatif tinggi.

2. DEFINISI BILANGAN PRIMA

Definisi bilangan prima:

1. Bilangan bulat positif p ($p > 1$) disebut bilangan prima jika pembaginya hanya 1 dan P.[1]
2. *Prime: a positive integer greater than 1with exactly two positive integer divisor.*[2]
3. *A **prime number** (or a **prime**) is a natural number which has exactly two **distinct** natural number divisors: 1 and itself.*[3]

Dapat disimpulkan bahwa bilangan prima adalah bilangan bulat positif lebih dari 1 dan hanya mempunyai dua pembagi, yaitu 1 dan bilangan tersebut. Sebagai contoh, 19 adalah bilangan prima karena 19 hanya habis dibagi oleh 19 dan 1. Hampir semua bilangan prima adalah bilangan ganjil dan angka 2 merupakan satu-satunya bilangan prima genap.

Dari definisi di atas, angka 1 tidak termasuk dalam bilangan prima, tetapi beberapa matematikawan pada abad XIX beranggapan bahwa angka 1 termasuk dalam himpunan bilangan prima yang akan dijelaskan lebih lanjut pada bab Sejarah Bilangan Prima. Bilangan-bilangan selain bilangan prima disebut bilangan komposit. Contohnya, 55 adalah bilangan komposit karena 55 habis dibagi 1, 5, 11, dan 55. Sampai saat ini, belum ditemukan pola yang pasti dalam distribusi bilangan prima sehingga cukup sulit untuk menemukan sebuah ataupun beberapa bilangan prima dalam suatu rentang. Cara yang paling mudah adalah menggunakan *Sieve of Eratosthenes.* Bilangan prima dapat disebut sebagai batu pembangun bilangan bulat positif. Hal ini dibuktikan dengan suatu teorema yang disebut **Teorema Fundamental Aritmetik (*The Fundamental Theorem of Arithmetic).***

> **Teorema Fundamental Aritmetik.** Setiap bilangan bulat positif yang lebih besar atau sama dengan 2 dapat dinyatakan sebagai perkalian satu atau lebih bilangan prima. [1]

3. SEJARAH BILANGAN PRIMA

Sejarah bilangan prima dimulai pada zaman Mesir kuno dengan ditemukannya sebuah catatan yang menyatakan penggunaan bilangan prima pada zaman tersebut. Namun, bilangan prima dan bilangan komposit pada zaman ini berbeda dengan bilangan prima dan bilangan komposit yang dikenal saat ini. Bukti lain permulaan sejarah bilangan prima adalah sebuah catatan penelitian bilangan prima oleh bangsa Yunani kuno. *Euclid's Elements* (300 BC) berisi beberapa teorema penting mengenai bilangan prima, termasuk ketakberhinggan bilangan prima dan teorema fundamental aritmetik. Euclid juga memperlihatkan bagaimana cara menyusun sebuah bilangan sempurna (*perfect number*) dari sebuah bilangan prima Mersenne yang ditemukan kemudian. Bukti lain adalah *Sieve of Eratosthenes*, yaitu sebuah cara untuk menghitung seluruh bilangan prima dalam suatu rentang tertentu.

Pada abad XVII, penelitian terhadap bilangan prima dilanjutkan kembali setelah berabad-abad berhenti. Pada tahun 1640, Pierre de Fermat memulainya dengan membuat Teorema Kecil Fermat (*Fermat's Little Theorem*) yang nantinya akan dibuktikan oleh Leibniz dan Euler. Kasus khusus dari teorema ini mungkin telah diketahui oleh bangsa Cina sebelumnya, namun belum ada bukti yang pasti mengenai hal ini. Lama setelah itu, Euler menemukan "lubang"pada teorema ini. Sebagai pengganti, seorang Prancis, Marin Mersenne, membuat suatu bentuk baru dari bilangan prima yang akhirnya namanya diabadikan menjadi nama bilangan ini, yaitu bilangan prima Mersenne (*Mersenne prime*). Cara penentuan inipun belum sempurna karena terdapat beberapa prima semu diantaranya.

Sampai abad XIX, banyak matematikawan masih beranggapan bahwa 1 adalah bilangan prima, dengan definisi bilangan prima adalah bilangan yang habis dibagi satu dan bilangan tersebut tanpa membatasi jumlah pembagi. Pada abad XIX, Legendre dan Gauss membuat sebuah konjektural untuk menghitung banyaknya bilangan prima yang kurang dari atau sama dengan suatu bilangan. Konjektural ini akhirnya dibuktikan pada tahun 1896 dan berganti nama menjadi Teorema Bilangan Prima (*Prime Number Theorem*). Sebelumnya, pada tahun 1859, Riemann mencoba membuktikan konjektural tersebut menggunakan fungsi-zeta.

Pencarian bilangan prima tidak berhenti sampai disitu, khususnya untuk bilangan-bilangan besar. Banyak matematikawan yang meneliti mengenai tes bilangan prima, contohnya: *Pepin's test* untuk bilangan Fermat (1877), *Lucas-Lehmer test* untuk bilangan Mersenne (1856), dan *Lucas-Lehmer test* yang digeneralisasikan.

Pada abad XX, penggunaan bilangan prima di luar bidang matematika mulai dikembangkan. Pada era 1970-an, ketika konsep kriptografi kunci-publik ditemukan, bilangan prima menjadi salah satu dasar pembuatan kunci algoritma enkripsi seperti RSA.

4. KETAKBERHINGGAAN BILANGAN PRIMA

Bilangan prima, sama halnya dengan bilangan bulat, mempunyai jumlah yang tak berhingga. Bukti dari pernyataan ini terdapat dalam *Euclid's Elements*. Euclid menyatakan bahwa terdapat lebih banyak bilangan prima daripada sejumlah berhingga bilangan prima yang diberikan[2]. Sebagai bukti, akan digunakan pembuktian terbalik atau membuktikan kontradiksi dari pernyataan tersebut. Diasumsikan ada sejumlah terbatas bilangan prima $p_1, p_2, p_3, \ldots p_n$. Lalu, dilakukan operasi sebagai berikut:

$$(1) Q = p_1 . p_2 \ldots p_n + 1$$

Didapat sebuah bilangan baru Q yang merupakan hasil perkalian seluruh bilangan prima ditambahkan dengan 1(satu). Menurut Teori Fundamental Aritmetik, seluruh bilangan bulat positif dapat difaktorkan menjadi satu atau lebih bilangan prima. Dengan dasar tersebut, Q dapat difaktorkan menjadi satu atau lebih bilangan prima. Namun, tidak ada satu pun bilangan prima (yang telah diasumsikan berjumlah berhingga) yang dapat habis membagi Q karena apapun bilangan primanya, misalkan p_j, Q dibagi p_j selalu akan menghasilkan sisa minimal 1. Jadi, terdapat suatu bilangan prima baru yang tidak termasuk dalam bilangan prima $p_1, p_2, p_3, \ldots p_n$, yaitu Q sendiri bila prima atau faktor prima dari Q. Kesimpulan ini kontradiktif dengan asumsi sebelumnya bahwa ada sejumlah berhingga bilangan prima. Oleh karena itu, bilangan prima berjumlah tak berhingga.

5. TEOREMA BILANGAN PRIMA (*PRIME NUMBER THEOREM*)

Pada bab sebelumnya telah dibuktikan bahwa banyaknya bilangan prima adalah tak berhingga. Secara umum, distribusi bilangan prima di ruang tak berhingga tersebut bersifat acak. Belum ada pola yang tepat untuk menggambarkan kemunculan bilangan prima dalam ruang tak berhingga. Namun, ada sebuah teorema yang dapat digunakan untuk menghitung banyaknya bilangan prima dalam suatu rentang tertentu.

> **Teorema Bilangan Prima (*Prime Number Theorem*).** Jumlah kemunculan bilangan prima yang kurang dari atau sama dengan suatu x [2]

Atau dapat ditulis:

$$(2)\ \pi(x) \sim {}^{x}/_{\ln x}$$

Dimana $\pi(x)$ adalah jumlah bilangan prima yang kurang dari sama dengan x.

Dengan menggunakan teorema tersebut, jumlah kemunculan bilangan prima yang kurang dari atau sama dengan x dapat dihitung secara manual. Namun, untuk nilai x yang cukup besar, nilai $\pi(x)$ dapat dihitung menggunakan sebuah algoritma secara cepat dan akurat dengan bantuan komputer. Contohnya: $\pi(100{,}000) = 9592$, and $\pi(10^{20}) = 2{,}220{,}819{,}602{,}560{,}918{,}840$. Untuk nilai yang lebih besar lagi, algoritma tersebut masih belum dapat menanganinya sehingga untuk menghitungnya dapat digunakan teorema di atas.

Selain menggunakan persamaan (2), nilai $\pi(x)$ dapat juga didekati dengan cara lain.

$$(3)\ \pi(x) \sim {}^{x}/_{(\ln x - 1)}$$

Dari Tabel 2 dapat dilihat bahwa pendekatan nilai $\pi(x)$ menggunakan persamaan (3) lebih akurat daripada persamaan (2).

Tabel 1: Tabel Nilai π(*x*)

x	π(*x*)
10	4
100	25
1,000	168
10,000	1,229
100,000	9,592
1,000,000	78,498
10,000,000	664,579
100,000,000	5,761,455
1,000,000,000	50,847,534

Tabel 2: Tabel Nilai π(*x*) dengan Berbagai Pendekatan

x	π(*x*)	$x/\ln x$	$x/(\ln x - 1)$
1,000	168	145	169
10,000	1,229	1,086	1,218
100,000	9,592	8,686	9,512
1,000,000	78,498	72,382	78,030
10,000,000	664,579	620,420	661,459
100,000,000	5,761,455	5,428,681	5,740,304

6. PENCARIAN BILANGAN PRIMA

5.1. Teorema Kecil Fermat

Teorema Kecil Fermat (*Fermat's Little Theorem*). Jika *p* adalah bilangan prima dan *a* adalah bilangan bulat yang tidak habis dibagi *p*, yaitu: $(4)\ a^p \equiv a \pmod p$, maka:

$(5)\ a^{p-1} \equiv 1 \pmod p$

Teorema ini mempunyai "lubang", yaitu terdapat bilangan komposit *n* yang memenuhi persamaan (5) bila *p* = *n*. Bilangan komposit ini disebut bilangan prima semu (*pseudoprime*).

Selain Teorema Kecil Fermat, Fermat membuat konjektur yang menyatakan bahwa:

$(6)\ F_n = 2^{2^n} + 1$

Dimana F_n adalah bilangan prima Fermat ke-*n*.

Konjektural ini telah dibuktikan sampai *n* = 4oleh Fermat. Namun, bilangan prima Fermat berikutnya merupakan bilangan komposit. Pembuktiannya dilakukan oleh Euler.

5.2. Bilangan Prima Mersenne

Bilangan Prima Mersenne ditemukan setelah bilangan prima Fermat ditemukan seolah menyempurnakannya. Bilangan prima Mersenne mempunyai rumus:

$(7)\ M_n = 2^p + 1$

Dengan p adalah bilangan prima.

5.3. Bilangan Prima Kembar (*Twin Primes*)

Bilangan prima kembar adalah dua bilangan prima dengan selisih 2, seperti 3 dan 5, 5 dan 7, serta 11 dan 13. Konjektural bilangan prima kembar menyatakan bahwa jumlah bilangan ini tak berhingga. Sampai saat ini, bilangan prima kembar yang telah ditemukan adalah $16{,}869{,}987{,}339{,}975 \cdot 2^{171{,}960} \pm 1$ angka dengan 51,779 digit.

5.4. *Sieve of Eratosthenes*

Sieve of Eratosthenes merupakan salah satu algoritma pencarian bilangan prima yang tertua. Walaupun umur algoritma ini sudah sangat tua, algoritma ini dianggap paling mudah diimplementasikan dalam komputasi. Algoritma *Sieve of Eratosthenes*:

1. Pertama-tama, buat sebuah *list* bilangan dari 2 sampai sebuah bilangan terbesar *n*.
2. Eliminasi/hilangkan dari *list* semua bilangan kelipatan 2.
3. Angka selanjutnya yang belum dihilangkan adalah bilangan prima.
4. Eliminasi/hilangkan dari *list* semua bilangan yang merupakan kelipatan dari angka pada langkah sebelumnya.
5. Ulangi langkah 3 dan langkah 4 sampai angka yang lebih besar dari $\sqrt{n}$.
6. Angka yang tersisa setelah langkah 5 adalah bilangan prima.

Algoritma ini mempunyai kompleksitas waktu O(n log log n). Versi algoritma ini dengan optimasi proses, seperti *wheel factorization*, mempunyai kompleksitas waktu O(n).

5.5. *Sieve of Atkin*

Sieve of Atkin adalah bentuk optimasi dari *Sieve of Eratosthenes*. Algoritma ini dibuat oleh A. O. L. Atkin dan Daniel J. Bernstein.

Algoritma *Sieve of Atkin*:

- Semua sisa bilangan adalah hasil modulo-60.
- Semua bilangan, termasuk *x* dan *y*, adalah bilangan bulat positif.
- Membalikkan sebuah entry dari daftar *sieve* berarti membalik penandaan keprimaan.\

1. Buat daftar hasil, diisi dengan 2, 3, dan 5.
2. Buat daftar *sieve* dengan sebuah entri untuk seluruh bilangan bulat positif. Semua entri tersebut ditandai sebagai bukan-prima.
3. Untuk setiap entri dalam daftar:
   - Jika entry tersebut untuk bilangan dengan sisa 1, 13, 17, 29, 37, 41, 49, atau 53, balikkan entri tersebut untuk setiap kemungkinan solusi $4x^2 + y^2 =$ *bilangan_entry*.
   - Jika entry tersebut untuk bilangan dengan sisa 7, 19, 31, atau 43, balikkan entri tersebut untuk setiap kemungkinan solusi $3x^2 + y^2 =$ *bilangan_entry*.

- Jika entry tersebut untuk bilangan dengan sisa 11, 23, 47, atau 59, balikkan entri tersebut untuk setiap kemungkinan solusi $3x^2 - y^2 = bilangan\_entry$ jika $x > y$.
- Jika entry tersebut mempunyai sisa yang lain, lewati entry tersebut.

4. Mulai dengan bilangan terkecil dalam daftar.
5. Maju ke bilangan setelahnya yang masih bertanda prima.
6. Masukkan bilangan tersebut ke daftar hasil
7. Kuadratkan bilangan tersebut dan tandai semua bilangan kelipatannya.
8. Ulangi langkah 5 sampai langkah 8.

Algoritma ini mempunyai kompleksitas waktu O(n / log log n). Dibandingkan dengan *Sieve of Eratosthenes*, algoritma ini sedikit lebih cepat.

7. APLIKASI BILANGAN PRIMA

Pada abad XX, penggunaan bilangan prima di luar bidang matematika mulai dikembangkan. Pada era 1970-an, ketika konsep kriptografi kunci-publik ditemukan, bilangan prima menjadi salah satu dasar pembuatan kunci algoritma enkripsi seperti RSA.

Algoritma RSA diperkenalkan oleh tiga peneliti MIT, yaitu Ron Rivest, Adi Shamir, dan Len Adleman. RSA menggunakan dasar bilangan prima dan aritmatika modulo dalam proses enkripsinya. Kunci enkripsi RSA tidak dirahasiakan dan dapat diketahui oleh umum, namun kunci deskripsinya bersifat rahasia. Untuk menemukan kunci dekripsi, seseorang harus memfaktorkan suatu bilangan komposit menjadi faktor primanya. Proses pemfaktoran ini bukan merupakan hal yang mudah sehingga tingkat keamanan enkripsi dengan RSA cukup tinggi.

Secara ringkas algoritma RSA adalah sebagai berikut:

1. Pilih dua buah bilangan prima sembarang, sebut *a* dan *b*, dirahasiakan.
2. Hitung $n = a \times b$, *n* tidak dirahasiakan.
3. Hitung $m = (a - 1) \times (b - 1)$. Setelah menghitung *m*, *a* dan *b* dapat dihapus.
4. Pilih sebuah bilangan bulat *e* yang relatif prima terhadap *m*.
5. Bangkitkan kunci dekripsi *d* dengan kekongruenan
$$(7)\ ed \equiv 1 \pmod m$$
6. Lakukan enkripsi dengan persamaan:
$$(8)\ c_i = p_i^{\,e} \bmod n$$
p_i adalah blok plainteks dan c_i adalah cipherteks yang diperoleh.
7. Proses dekripsi dilakukan dengan persamaan:
$$(8)\ p_i = c_i^{\,d} \bmod n$$

DAFTAR REFERENSI

[1] Munir, Rinaldi. 2008. *Diktat Kuliah IF2091 Struktur Diskrit*. Bandung: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. Hal. V-19 – V-21.

[2] Rosen, Kenneth H. 2007. *Discrete Mathematics and Its Application, Sixth Edition*. New York: McGraw-Hill.

[3] Prime number – Wikipedia, the free encyclopedia. http://en.wikipedia.org/Prime_Number.html. Tanggal akses: 22 Desember 2008 pk. 12.00

[4] How many primes are there? Tanggal akses: 04 Januari 2009 pk. 15.00

[5] How to find primes and prove primality (merged version). Tanggal akses: 04 Januari 2009 pk. 15.00

[6] Sieve of Eratosthenes – Wikipedia, the free encyclopedia. http://en.wikipedia.org/Sieve_of_Eratosthenes.html. Tanggal akses: 04 Januari 2009 pk. 15.00

[7] RSA – Wikipedia, the free encyclopedia. http://en.wikipedia.org/RSA.html. Tanggal akses: 04 Januari 2009 pk. 15.00

[8] The Largest Known prime by Year_ A Brief History. Tanggal akses: 04 Januari 2009 pk. 15.00

Tabel 2: Bilangan Prima Mersenne Terbesar

Number	Digits	Year	Machine	Prover
$180(M_{127})^2+1$	79	1951	EDSAC1	Miller & Wheeler
M_{521}	157	1952	SWAC	Robinson (Jan 30)
M_{607}	183	1952	SWAC	Robinson (Jan 30)
M_{1279}	386	1952	SWAC	Robinson (June 25)
M_{2203}	664	1952	SWAC	Robinson (Oct 7)
M_{2281}	687	1952	SWAC	Robinson (Oct 9)
M_{3217}	969	1957	BESK	Riesel
M_{4423}	1,332	1961	IBM7090	Hurwitz
M_{9689}	2,917	1963	ILLIAC 2	Gillies
M_{9941}	2,993	1963	ILLIAC 2	Gillies
M_{11213}	3,376	1963	ILLIAC 2	Gillies
M_{19937}	6,002	1971	IBM360/91	Tuckerman
M_{21701}	6,533	1978	CDC Cyber 174	Noll & Nickel
M_{23209}	6,987	1979	CDC Cyber 174	Noll
M_{44497}	13,395	1979	Cray 1	Nelson & Slowinski
M_{86243}	25,962	1982	Cray 1	Slowinski
M_{132049}	39,751	1983	Cray X-MP	Slowinski
M_{216091}	65,050	1985	Cray X-MP/24	Slowinski
$391581*2^{216193}-1$	65,087	1989	Amdahl 1200	Amdahl Six
M_{756839}	227,832	1992	Cray-2	Slowinski & Gage et al. (notes)
M_{859433}	258,716	1994	Cray C90	Slowinski & Gage
$M_{1257787}$	378,632	1996	Cray T94	Slowinski & Gage
$M_{1398269}$	420,921	1996	Pentium (90 Mhz)	Armengaud, Woltman, et al. [GIMPS]
$M_{2976221}$	895,932	1997	Pentium (100 Mhz)	Spence, Woltman, et al. [GIMPS]
$M_{3021377}$	909,526	1998	Pentium (200 Mhz)	Clarkson, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{6972593}$	2,098,960	1999	Pentium (350 Mhz)	Hajratwala, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{13466917}$	4,053,946	2001	AMD T-Bird (800 Mhz)	Cameron, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{20996011}$	6,320,430	2003	Pentium (2 GHz)	Shafer, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{24036583}$	7,235,733	2004	Pentium 4 (2.4GHz)	Findley, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{25964951}$	7,816,230	2005	Pentium 4 (2.4GHz)	Nowak, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{30402457}$	9,152,052	2005	Pentium 4 (2GHz upgraded to 3GHz)	Cooper, Boone, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{32582657}$	9,808,358	2006	Pentium 4 (3 GHz)	Cooper, Boone, Woltman, Kurowski, et al. [GIMPS, PrimeNet]
$M_{43112609}$	12,978,189	2008	Intel Core 2 Duo E6600 CPU (2.4 GHz)	E_Smith, Woltman, Kurowski, et al. [GIMPS, PrimeNet]