Teori Bilangan di Balik GIMPS: Misi Mencari Bilangan Prima Mersenne Terbesar yang Diketahui

Akeyla Pradia Naufal 13519178[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13519178@std.stei.itb.ac.id

***Abstract*—GIMPS adalah sebuah proyek kolosal yang bertujuan untuk mencari bilangan prima Mersenne terbesar yang diketahui. Dalam makalah ini, akan dibahas mengenai hukum matematika (modulo dan bilangan Mersenne) yang digunakan sebagai dasar algoritma pencarian bilangan prima pada GIMPS dan bagaimana cara GIMPS bekerja.**

***Keywords*—prima Mersenne, GIMPS, modulo, bilangan Mersenne**

I. PENDAHULUAN

Bilangan prima adalah bilangan asli p yang hanya memiliki dua buah faktor positif yakni 1 dan p. Euclid, dengan menggunakan metode kontradiksi, telah membuktikan bahwa banyak bilangan prima adalah tak hingga. Sehingga, pencarian bilangan prima terbesar, secara teori, tidak akan pernah berakhir.

Bilangan prima sangat penting dalam ilmu teori bilangan dikarenakan sifatnya yang menjadi dasar dari setiap bilangan asli. Hal ini ditunjukkan dengan Teorema Fundamental Aritmetika yang menyatakan bahwa setiap bilangan asli yang lebih besar dari satu dapat difaktorkan menjadi perkalian beberapa bilangan prima secara unik.

Mengalikan dua buah bilangan memiliki kompleksitas yang jauh lebih rendah daripada mencari faktorisasi prima sebuah bilangan asli. Hal ini membuat perkalian dua bilangan prima yang besar akan sangat susah untuk difaktorkan kembali. Fakta ini digunakan dalam beberapa algoritma kriptografi seperti RSA. Hal ini juga membuat pengetahuan akan bilangan-bilangan prima yang cukup besar menjadi hal yang penting dalam keamanan komputer yang menggunakan RSA.

GIMPS (*Great Internet Mersenne Prime Search*) adalah sebuah proyek kolaboratif di internet untuk mencari bilangan prima bertipe Mersenne. Proyek ini dimulai sejak tahun 1996 dan proyek ini telah menemukan tujuh belas bilangan prima Mersenne. Sebanyak lima belas di antaranya saat ditemukan merupakan bilangan prima terbesar yang diketahui. Hingga tahun 2020, bilangan prima terbesar yang diketahui (yang juga ditemukan oleh GIMPS) adalah $2^{82589933} - 1$ yang memiliki 24.862.048 digit pada representasi desimalnya.

Bilangan prima dengan digit sebanyak ini begitu besar sehingga belum ada kebutuhan untuk menggunakan bilangan prima ini di sistem kriptografi. Kurangnya kebutuhan bilangan prima yang sangat besar ini bukanlah sebuah masalah. Tujuan utama dari pencarian bilangan prima terbesar yang diketahui ini, seperti ditulis di halaman web GIMPS itu sendiri, adalah untuk mencatatkan sejarah baru dalam pencarian bilangan prima Mersenne itu sendiri.

II. LANDASAN TEORI

A. Modulo

Diberikan sebuah bilangan bulat $a, b,$ dan m dengan $m \neq 0$. Bilangan a disebut *kongruen* dengan b modulo m apabila m membagi $b - a$. Hal ini sering dinotasikan dengan

$$a \equiv b \pmod{m}$$

Beberapa properti yang berlaku pada modulo adalah:

- Untuk setiap bilangan bulat a, $a \equiv a \pmod{n}$
- Untuk setiap bilangan asli a, b, c dan d, jika $a \equiv b \pmod{n}$ dan $b \equiv c \pmod{n}$, maka $a \equiv c \pmod{n}$
- Untuk setiap bilangan asli a, b, c dan d, jika $a \equiv b \pmod{n}$, maka $b \equiv a \pmod{n}$
- Untuk setiap bilangan asli a, b, c dan d, jika $a \equiv b \pmod{n}$, maka untuk sebarang bilangan bulat k, berlaku $ka \equiv kb \pmod{n}$
- Untuk setiap bilangan asli a, b, c dan d, jika $a \equiv b \pmod{n}$ dan $c \equiv d \pmod{n}$, maka untuk sebarang bilangan bulat j, k, berlaku $ja + kc \equiv jb + kd \pmod{n}$
- Untuk setiap bilangan asli a, b, c dan d, jika $a \equiv b \pmod{n}$ dan $c \equiv d \pmod{n}$, maka $ac \equiv bd \pmod{n}$
- Untuk setiap bilangan asli $a, b,$ dan x, jika x relatif prima dengan n dan $xa \equiv xb \pmod{n}$, maka $a \equiv b \pmod{n}$

B. Teorema Fermat Kecil

Teorema Fermat Kecil mengatakan bahwa untuk setiap bilangan prima p dan bilangan asli a yang relatif prima dengan a, maka

$$a^{p-1} \equiv 1 \pmod{p}$$

Bukti:

Perhatikanlah bahwa $a, 2a, \ldots, (p-1)a$ semuanya akan relatif prima dengan p. Akan dibuktikan bahwa semua bilangan $a, 2a, \ldots, (p-1)a$ memiliki sisa yang berbeda saat dibagi oleh p. Asumsikan ada dua bilangan berbeda yang memiliki sisa yang sama saat dibagi oleh p yakni ia dan ja dengan $1 \leq i < j \leq p-1$. Maka, $ia \equiv ja (\mathrm{mod}\ p)$.

Berdasarkan definisi, p membagi $ia - ja = a(i-j)$. Karena a dan p relatif prima, maka p tidak membagi a. Akibatnya, p harus membagi $i-j$. Namun, $1 \leq i < j \leq p-1$ sehingga $p-2 > j-i > 1$. Tidak ada bilangan asli di antara 1 dan $p-2$ yang habis dibagi oleh p. Jadi, asumsi bahwa ada dua bilangan berbeda yang sisa bagi saat dibagi oleh p sama, salah. Jadi, $a, 2a, 3a, \ldots, (p-1)a$ memiliki sisa yang berbeda saat dibagi oleh p. Karena sisa bagi saat sebuah bilangan yang relatif prima dengan p dibagi oleh p adalah $1, 2, 3, \ldots$, atau $p-1$, sisa bagi dari $1a, 2a, 3a, \ldots, (p-1)a$ saat dibagi oleh p mreupakan permutasi dari $1, 2, \ldots, p-1$. Sehingga

$$a \cdot 2a \cdot \ldots \cdot (p-1)a \equiv a \cdot 2a \cdot \ldots \cdot (p-1)a (\mathrm{mod}\ p)$$
$$(p-1)!\, a^{p-1} \equiv (p-1)! \ (\mathrm{mod}\ p)$$

Karena $(p-1)!$ merupakan hasil kali semua bilangan asli yang lebih kecil dari p, maka $(p-1)!$ relatif prima dengan p. Sehingga faktor $(p-1)!$ di kedua ruas dapat dihapus, menyisakan

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

Perlu diperhatikan bahwa invers dari teorema ini tidaklah berlaku. Artinya, jika a dan p adalah bilangan asli sehingga $a^{p-1} \equiv 1 (\mathrm{mod}\ p)$, maka p belum tentu merupakan bilangan prima.

C. Order Modulo

Suatu bilangan asli z disebut sebagai order dari bilangan asli a dalam modulo bilangan asli m apabila a relatif prima dengan m, $a^z \equiv 1 \ (\mathrm{mod}\ m)$, dan z adalah bilangan asli terkecil yang memiliki sifat ini.

Terdapat sebuah properti yang menarik yang berhubungan dengan order modulo yakni apabila $a^k \equiv 1 \ (\mathrm{mod}\ m)$ dan z adalah order dari a modulo m maka z membagi k.

Bukti:

Berdasarkan definisi, z adalah bilangan asli terkecil yang memenuhi $a^z \equiv (\mathrm{mod}\ m)$ sehingga k harus tidak boleh lebih kecil dari z. Bilangan k dapat dituliskan secara unik dalam bentuk $k = qz + r$ dengan q adalah bilangan asli dan r adalah bilangan cacah r yang memenuhi $0 \leq r < z$. Sehingga

$$a^k \equiv 1 \ (\mathrm{mod}\ m)$$
$$a^{qz+r} \equiv 1 \ (\mathrm{mod}\ \ m)$$
$$(a^z)^q \cdot a^r \equiv 1 \ (\mathrm{mod}\ m)$$
$$a^r \equiv 1 \ (\mathrm{mod}\ m)$$

Namun, karena $r < z$ dan berdasarkan definisi, z adalah bilangan asli terkecil yang memenuhi $a^z \equiv 1 \ (\mathrm{mod}\ m)$, maka dapat disimpulkan bahwa nilai r yang mungkin adalah $r = 0$. Sehingga, $k = qz$ dan akibatnya, z habis membagi k. ■

D. Residu Kuadratik dan Notasi Legendre

Suatu bilangan bulat a disebut sebagai residu kuadratik pada modulo bilangan asli n apabila terdapat sebuah bilangan bulat x sehingga

$$x^2 \equiv a \ (\mathrm{mod}\ n).$$

Jika tidak ada nilai x yang memenuhi maka a disebut sebagai nonresidu kuadratik modulo n.

Banyak bilangan asli yang lebih kecil dari bilangan prima ganjil p dan merupakan residu kuadratik modulo p adalah tepat $\frac{p-1}{2}$. Hal ini dibuktikan di bawah ini.

Bukti:

Misalkan $0 < i < p$ adalah bilangan asli. Perhatikanlah bahwa

$$(p-i)^2 = p^2 - 2pi + i^2$$
$$\equiv i^2 \ (\mathrm{mod}\ p).$$

Selain itu, jika $0 < m < n < p$ dengan m dan n adalah dua bilangan asli dengan $m + n \neq p$, maka

$$m^2 - n^2 = (m-n)(m+n)$$

tidak mungkin habis dibagi oleh p.

Jadi, untuk nilai $i = 1, 2, \ldots, \frac{p-1}{2}$, nilai i^2 akan berbeda dalam modulo p dan sudah memuat semua residu kuadratik yang mungkin dalam modulo p. ■

Misalkan p adalah bilangan prima ganjil dan a serta b adalah bilangan yang relatif prima dengan p.

1. Jika a dan b keduanya adalah residu kuadratik modulo p, maka ab juga adalah residu kuadratik modulo p
2. Jika a dan b keduanya adalah nonresidu kuadratik modulo p, maka ab adalah residu kuadratik modulo p.
3. Jika salah satu dari a dan b adalah residu kuadratik modulo p dan yang lainnya merupakan nonresidu kuadratik modulo p, maka ab adalah nonresidu kuadratik modulo p.

Bukti:

Jika a dan b adalah dua residu kuadratik modulo n, maka ada bilangan bulat x dan y sehingga $x^2 \equiv a \ (\mathrm{mod}\ n)$ dan $y^2 \equiv b \ (\mathrm{mod}\ n)$. Jelas bahwa $ab \equiv (xy)^2 \ (\mathrm{mod}\ n)$. Jadi, ab juga merupakan residu kuadrat modulo n.

Perhatikanlah bahwa di antara $i = 1, 2, \ldots, p-1$, ada tepat setengahnya yang merupakan residu kuadratik modulo p. Setengah sisanya merupakan nonresidu kuadratik modulo p.

Perhatikan pula bahwa untuk setiap pasangan bilangan asli i dan j dengan $0 < i < j < p$ dan bilangan bulat a yang relatif prima dengan p, nilai ai dan aj akan berbeda dalam modulo p. Hal ini dapat dibuktikan dengan fakta bahwa $ai - aj = a(i-j)$ tidak mungkin habis dibagi oleh p.

Fakta pada dua paragraf sebelumnya dapat digunakan untuk menyimpulkan bahwa $a, 2a, 3a, \ldots, (p-1)a$ memuat tepat $\frac{p-1}{2}$ residu kuadratik modulo p. Jika a adalah residu kuadratik

modulo p, ke-$\frac{p-1}{2}$ residu kuadratik modulo p ini hanya tercapai saat i juga merupakan residu kuadratik modulo p. Akibatnya, ai adalah nonresidu kuadratik modulo p saat i merupakan nonresidu kuadratik modulo p dan a adalah residu kuadratik modulo p.

Dengan argumen yang serupa, jika a adalah nonresidu kuadratik modulo p, ai adalah residu kuadratik modulo p saat i adalah nonresidu kuadratik modulo p dan nonresidu kuadratik saat i adalah residu kuadratik modulo p. ■

Gauss menotasikan a adalah residu kuadratik modulo b dengan $a \,\mathrm{R}\, b$ dan $a \,\mathrm{N}\, b$ jika a adalah nonresidu kuadratik modulo b. Dikenal pula simbol Legendre, yakni jika a adalah bilangan bulat dan p adalah bilangan prima ganjil, maka

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{jika } p \text{ membagi } a \\ 1, & \text{jika } p \nmid a \text{ dan } a \,\mathrm{R}\, p \\ -1, & \text{jika } p \mid a \text{ dan } a \,\mathrm{N}\, p \end{cases}$$

Properti sebelumnya dapat diterjemahkan dalam simbol Legendre sebagai

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$$

Kriteria Euler dapat membantu menentukan apakah nilai dari $\left(\frac{a}{p}\right)$. Kriteria Euler menyatakan bahwa:

Jika p adalah bilangan prima ganjil dan a adalah bilangan asli yang tidak habis dibagi p, maka

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

Bukti:

Jika $\left(\frac{a}{p}\right) = 1$, a adalah residu kuadratik modulo p dan berdasarkan definisi, pasti terdapat sebuah bilangan asli i sehingga $i^2 \equiv a \pmod p$. i pastilah merupakan bilangan yang realtif prima dengan p. Oleh karena itu, berdasarkan Teorema Fermat Kecil, $i^{p-1} \equiv 1 \pmod p$. Maka, $a^{\frac{p-1}{2}} \equiv 1 \pmod p$.

Untuk kasus saat $\left(\frac{a}{p}\right) = -1$, perhatikanlah bahwa solusi dari persamaan $x^{\frac{p-1}{2}} \equiv 1 \pmod p$ untuk $x = 1, 2, \ldots, p-1$ adalah saat x merupakan residu kuadratik modulo p. Perhatikanlah bahwa p haruslah membagi $x^{p-1} - 1 = \left(x^{\frac{p-1}{2}} - 1\right)\left(x^{\frac{p-1}{2}} + 1\right)$. Jika x residu kuadratik modulo p, maka $x^{\frac{p-1}{2}} - 1$ pasti habis dibagi oleh p. Jika x bukan residu kuadratik, haruslah p habis membagi $x^{\frac{p-1}{2}} + 1$. Hal ini ekivalen dengan $x^{\frac{p-1}{2}} \equiv -1 \equiv \left(\frac{a}{p}\right) \pmod p$ untuk x nonresidu kuadratik modulo p. ■

Gauss juga memberi sebuah lemma yang dapat menentukan apakah suatu bilangan asli a merupakan residu kuadratik dalam modulo bilangan prima p.

Lemma:

Jika a adalah bilangan asli yang tidak habis dibagi oleh bilangan prima p dan untuk setiap $k = 1, 2, \ldots, \frac{p-1}{2}$,

$$ka = pq_k + r_k$$

dengan q_k dan r_k adalah bilangan cacah dengan $0 \le r_k < p$. ■

Misalkan $b_1, b_2, \ldots, b_m$ adalah banyak nilai r_i yang berbeda dan bernilai kurang dari $\frac{p}{2}$. Misalkan pula $c_1, c_2, \ldots, c_n$ adalah suku-suku r_i yang tersisa. Maka,

$$\left(\frac{a}{p}\right) = (-1)^n$$

Bukti:

Perhatikanlah bahwa

$$\prod_{i=1}^{m} b_i \prod_{j=1}^{n} c_j = \prod_{k=1}^{\frac{p-1}{2}} r_k = \prod_{k=1}^{\frac{p-1}{2}} (ka - pq_k) \equiv \prod_{k=1}^{\frac{p-1}{2}} ka \pmod p \equiv a^{\frac{p-1}{2}} \left(\frac{p-1}{2}\right)! \pmod p$$

Karena $\frac{p}{2} < c_j \le p-1, j = 1, \ldots, n$, maka $1 \le p - c_j \le \frac{p-1}{2}$. Perhatikan pula bahwa tidak mungkin ada bilangan asli i dan j sehingga $c_j + b_i = p$. Hal ini dikarenakan apabila $p = c_j + b_i$ maka $p = c_j + b_i = ay - pq_y + az - pq_z$ untuk suatu bilangan asli $1 \le y, z \le \frac{p-1}{2}$. Ini tidak mungkin karena akan mengakibatkan p membagi $y + z$ padahal $2 \le y + z \le p - 1$.

Sehingga, dapat disimpulkan bahwa bilangan-bilangan asli $b_1, \ldots, b_m, p - c_1, \ldots, p - c_n$ semuanya berbeda dan semuanya membentuk bilangan asli dari 1 sampai $\frac{p-1}{2}$. Akibatnya,

$$\prod_{i=1}^{m} b_i \prod_{j=1}^{n} (p - c_j) \equiv \left(\frac{p-1}{2}\right)!.$$

$$(-1)^n \prod_{i=1}^{m} b_i \prod_{j=1}^{n} c_j \equiv \left(\frac{p-1}{2}\right)! \pmod p$$

$$(-1)^n a^{\frac{p-1}{2}} \left(\frac{p-1}{2}\right)! \equiv \left(\frac{p-1}{2}\right)! \pmod p.$$

Karena $\left(\frac{p-1}{2}\right)!$ relatif prima dengan p, dapat disimpulkan bahwa $a^{\frac{p-1}{2}} \equiv (-1)^n \pmod p$. Lemma terbukti dengan menggunakan kriteria Euler. ■

Dengan Lemma Gauss ini, dapat dibuktikan bahwa

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$$

Bukti:

Pada lemma Gauss, substitusikan $a = 2$. Maka, $ka \leq (p-1)$ dan akibatnya, nilai q_k akan selalu sama dengan 0 dan $r_i = 2i$ untuk $i = 1,2, \ldots, \frac{p-1}{2}$.

Banyak nilai r_i yang lebih kecil dari $\frac{p}{2}$ adalah $\left\lfloor \frac{p}{4} \right\rfloor$. Banyak nilai r_i yang tersisa adalah $n = \frac{p-1}{2} - \left\lfloor \frac{p}{4} \right\rfloor$. Karena p adalah bilangan prima ganjil, maka $p = 4x + 1$ atau $p = 4x + 3$ untuk sebuah bilangan cacah x.

Jika $p = 4x + 1$, maka $n = 2x - x = x$ dan $\frac{p^2-1}{8} = 2x^2 + x$. Paritas dari $\frac{p^2-1}{8}$ dan n sama. Sehingga, $(-1)^{\frac{p^2-1}{8}} = (-1)^n = \left(\frac{2}{p}\right)$.

Jika $p = 4x + 3$, maka $n = 2x + 1 - x = x + 1$ dan $\frac{p^2-1}{8} = 2x^2 + 3x + 1$. Paritas dari $\frac{p^2-1}{8}$ dan n sama. Sehingga, $(-1)^{\frac{p^2-1}{8}} = (-1)^n = \left(\frac{2}{p}\right)$. Jadi, dapat disimpulkan bahwa $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$. ■

E. Bilangan *Smooth* dan Bilangan *Powersmooth*

Untuk suatu bilangan asli B dan n, n disebut sebagai B-*smooth* apabila tidak faktor prima dari n yang lebih besar dari B. Selain itu, n disebut sebagai B-*powersmooth* apabila untuk setiap faktor prima-berpangkat dari n, tidak ada yang nilainya melebihi B. Dengan kata lain, jika p adalah bilangan prima, k adalah bilangan asli, serta p^k habis membagi n, maka $p^k \leq B$. Bilangan *powersmooth* akan digunakan dalam algoritma Pengetesan Bilangan Prima Pollard $p - 1$.

F. Bilangan Prima Mersenne

Bilangan Mersenne adalah bilangan asli yang berbentuk $M_n = 2^n - 1$ dengan n adalah bilangan asli.

Semua bilangan Mersenne yang merupakan bilangan prima haruslah memeiliki eksponen yang merupakan bilangan prima. Dengan kata lain, jika $M_p = 2^p - 1$ adalah bilangan prima, maka p adalah bilangan prima.

Bukti:

Asumsikan $M_p = 2^p - 1$ adalah bilangan prima tetapi p bukan bilangan prima. Sehingga, p adalah bilangan komposit dan dapat ditulis sebagai $p = qr$ dengan q dan r adalah bilangan asli yang lebih besar dari 1. Sehingga, $M_p = 2^p - 1$

$$= 2^{qr} - 1$$
$$= (2^q - 1)(2^{qr-q} + 2^{qr-2q} + \cdots + 2^q + 1)$$

Karena $q > 1$, nilai dari $2^q - 1$ akan lebih besar dari 1. Demikian juga, karena $r > 1$, nilai dari 2^{qr-r} akan lebih besar dari 1 dan $(2^{qr-q} + 2^{qr-2q} + \cdots + 2^q + 1)$ juga akan lebih besar dari 1. Sebagai akibatnya, M_p merupakan hasil kali dua buah bilangan asli yang lebih besar dari 1. Jadi, M_p bukan merupakan bilangan prima. Karena terjadi kontradiksi, asumsi bahwa p bukan bilangan prima adalah salah dan dapat disimpulkan bahwa p adalah bilangan prima. ■

Kebalikannya tidak berlaku. Jika p adalah bilangan prima maka $M_p = 2^p - 1$ belum tentu merupakan bilangan prima. Sebagai contoh, 11 adalah bilangan prima tetapi $M_{11} = 2^{11} - 1 = 2047$ adalah bilangan komposit karena $2047 = 23 \times 89$.

Terdapat beberapa properti dari bilangan prima Mersenne yang dapat digunakan untuk mengecek faktor dari bilangan prima Mersenne.

Properti 1.

Semua faktor prima dari bilangan Mersenne $M_p = 2^p - 1$ berbentuk $n = 2kp + 1$ dengan k adalah bilangan cacah.

Bukti:

Nilai p tentu saja tidak mungkin sama dengan 2. Sehingga, p adalah bilangan ganjil. Jika q adalah sebuah bilangan prima yang membagi $2^p - 1$, maka $2^p \equiv 1 (\mathrm{mod}\, q)$. Selain itu, dengan menggunakan Teorema Fermat Kecil, $2^{q-1} \equiv 1 \pmod{q}$. Jika z adalah order dari 2 dalam modulo q, maka $2^z \equiv 1 (\mathrm{mod}\, q)$ dan z akan membagi p dan $q - 1$ secara sekaligus. Karena 2^1 tidak mungkin kongruen dengan 1 dalam modulo q, nilai z yang mungkin adalah $z = p$. Karena z membagi $q - 1$ dan $z = p$, dapat disimpulkan bahwa p akan membagi $q - 1$. Selain itu, karena q adalah bilangan prima ganjil, maka $q - 1$ adalah bilangan genap. Dengan menggabungkan kedua fakta ini, dapat disimpulkan bahwa $q = 2kp + 1$ untuk suatu bilangan cacah k. ■

Properti 2.

Dua bilangan Mersenne yang eksponennya relatif prima akan relatif prima.

Bukti:

Misalkan $M_x = 2^x - 1$ dan $M_y = 2^y - 1$ adalah dua buah bilangan Mersenne dengan x relatif prima dengan y. Asumsikan bahwa M_x dan M_y tidak relatif prima. Maka akan terdapat sebuah bilangan prima p yang membagi keduanya. Sehingga, $2^x \equiv 1 \pmod{p}$ dan $2^y \equiv 1 \pmod{p}$. Misalkan z adalah order dari 2 dalam modulo p, maka z akan membagi x dan z akan membagi y juga. Akibatnya, z adalah faktor bersama dari x dan y. Hal ini kontradiktif dengan fakta bahwa x dan y relatif prima. Jadi, dapat disimpulkan bahwa M_x dan M_y relatif prima. ■

Properti 3.

Setiap faktor prima dari bilangan prima Mersenne akan konguren dengan 1 atau -1 dalam modulo 8.

Bukti:

Jika q adalah faktor prima dari $M_p = 2^p - 1$, maka $2^{p+1} \equiv 2 \pmod{q}$. Karena $p + 1$ adalah bilangan genap, maka 2 adalah residu kuadratik modulo q. Padahal, sebelumnya sudah diketahui bahwa $\left(\frac{2}{q}\right) =$

$(-1)^{\frac{q^2-1}{8}}$. Jadi, dapat disimpulkan bahwa $\frac{q^2-1}{8}$ adalah bilangan genap. Hal ini tercapai hanya saat $q \equiv 1$ atau $-1 (\mathrm{mod}\ q)$. ■

Sampai makalah ini dibuat, belum diketahui pasti apakah banyak bilangan prima Mersenne tak hingga atau tidak. Faktanya, per 2020, baru 53 bilangan prima Mersenne yang ditemukan.

III. Pencarian Bilangan Prima Mersenne Oleh GIMPS

A. GIMPS

GIMPS (*Great Internet Mersenne Prime Search*) adalah sebuah proyek kolaborasi pencarian bilangan prima Mersenne menggunakan software Prime95 (Windows, MacOS) atau Mprime (Linux). GIMPS ditemukan oleh George Woltman pada tahun 1996. Hingga 2020, proyek GIMPS sudah menemukan dan memverifikasi 17 buah bilangan prima Mersenne; bilangan prima Mersenne yang ditemukan juga masih memegang rekor sebagai bilangan prima terbesar yang diketahui manusia. Informasi tentang GIMPS dapat dilihat pada situs resminya, https:/mersenne.org.

GIMPS menggunakan sistem pendistribusian perhitungan melalui server PrimeNet yang ditemukan oleh Scott Kurowski pada tahun 1997. Program berkomunikasi dengan server dengan protokol HTTP. Program ini hanya mengirim beberapa ratus byte per minggu dan tidak memerlukan koneksi internet secara terus menerus. Server PrimeNet akan memberikan tugas yang berbeda untuk setiap komputer. Tugas ini dapat dibagi menjadi tiga jenis:

1. Pengetesan bilangan prima perdana.
   Tugas ini akan diberikan kepada komputer dengan kecepatan CPU 1,4 GHz atau lebih. Pengetesan ini akan mengecek apakah sebuah bilangan Mersenne merupakan bilangan prima atau tidak
2. Pemverifikasian keprimaan sebuah bilangan
   Tugas ini akan diberikan kepada komputer dengan kecepatan CPU 1 – 1,4 GHz. Tugas ini akan memverifikasi kembali apakah sebuah bilangan Mersenne benar-benar adalah bilangan prima atau komposit.
3. Pemfaktoran bilangan
   Tugas ini akan diberikan kepada komputer dengan kecepatan CPU kurang dari 1 GHz. Tugas ini akan membantu mengeliminasi kemungkinan keprimaan beberapa bilangan Mersenne.

Dalam melakukan tugas-tugas ini, program Prime95 akan menggunakan algoritma pemfaktoran dan pengetesan bilangan prima berikut:

1. Algoritma Pemfaktoran Trial
2. Algoritma Pemfaktoran Pollard $p - 1$
3. Algoritma Pengetesan Bilangan Prima Lucas-Lehmer
4. Algoritma Pengetesan Bilangan Prima Fermat
5. Algoritma Pemfaktoran Kurva Eliptik (ECM)

Guna menghindari adanya kesalahan perhitungan saat menentukan sifat keprimaan sebuah bilangan Mersenne, GIMPS mewajibkan setiap bilangan Mersenne yang sudah diperiksa sifat keprimaannya satu kali untuk diverifikasi kembali oleh pengguna yang berbeda. Pemverifikasian ini disebut sebagai *double-check.*

Jika sebuah bilangan Mersenne diketahui sudah terverifikasi sebagai bilangan prima, maka pengguna yang pertama kali menemukan bahwa bilangan tersebut prima akan mendapatkan hadiah uang sebesar $3000. Hadiah ini akan menjadi jauh lebih besar, yaitu $50000, apabila bilangan prima Mersenne yang ditemukan memiliki seratus juta digit dalam representasi desimal.

Jika sebuah bilangan Mersenne telah terverifikasi sebagai bilangan komposit melalui Algoritma Lucas-Lehmer atau Algoritma Fermat, pengguna dapat mencoba mencari faktor-faktor prima dari bilangan Mersenne ini. Algoritma yang digunakan adalah Algoritma Trial, Algoritma $p - 1$, dan Algoritma ECM.

Pembagian tugas perhitungan di GIMPS dapat dilihat pada gambar 3.1.

Project	WorkType	Description	Notes
GIMPS	TF-LMH	trial factoring LMH	screen for not-prime, factoring to low limits
GIMPS	PM1-R	factor P-1 redo	screen for not-prime, reassigned as P-factoring
GIMPS	TF	trial factoring	screen for not-prime, assigned as factoring
GIMPS	PM1-S	factor P-1 small	screen for not-prime, assigned as factoring
GIMPS	PM1-L	factor P-1 large	screen for not-prime, assigned as P-factoring
GIMPS	ECM	factor ECM small	screen for not-prime, assigned as ECM
GIMPS	PRP-D	PRP double-check	probable prime test, assigned as PRP verification
GIMPS	PRP-CF	PRP cofactor test	probable prime test, assigned as PRP cofactor test
GIMPS	LL	LL first test	primality test, assigned as first LL test
GIMPS	D	LL double-check	primality test, assigned as LL verification
GIMPS	PRP-CF-D	PRP cofactor test double-check	probable prime test, assigned as PRP cofactor verification
GIMPS	ECM-F	ECM Fermat	ECM Fermat test, assigned for ECM Fermat testing
GIMPS	ECM-MF	ECM more factors	ECM more factors, assigned for more ECM factoring
GIMPS	PRP	PRP test	probable prime test, assigned as PRP test
GIMPS	ECM-C	ECM Cunningham	ECM Cunningham, assigned for ECM Cunningham testing
GIMPS	CERT	PRP proof certification	proof verification, assigned to reliable computers

Gambar 3.1. Pembagian tugas pehitungan di GIMPS
https://www.mersenne.org/worktypes/

B. Algoritma Pemfaktoran Trial

Algoritma Pemfaktoran Trial adalah algoritma pemfaktoran yang paling sederhana. Algoritma ini dapat digunakan untuk menentukan apakah sebuah bilangan memiliki faktor prima yang relatif kecil sehingga digunakan untuk mengeliminasi kemungkinan sebuah bilangan Mersenne merupakan bilangan prima.

Pada bentuk paling dasarnya, algoritma pemfaktoran ini bekerja sebagai berikut:

1. Tentukan bilangan yang ingin dicari faktornya, misalkan n.
2. Cari semua bilangan prima yang nilainya tidak melebihi $\sqrt{n}$.
3. Lakukan pengecekan apakah di antara seluruh bilangan prima ini ada yang habis membagi n

Untuk mencari bilangan prima yang tidak melebihi $\sqrt{n}$, dapat digunakan *sieve of Eratosthenes* atau variasinya.

Algoritma Pemfaktoran Trial yang ada di Prime95 sudah divariasikan agar bekerja lebih cepat terhadap bilangan Mersenne $M_p = 2^p - 1$ dengan p adalah bilangan prima. Beberapa variasi yang dilakukan adalah:

1. *Sieve of Eratosthenes* yang digunakan divariasikan sehingga hanya menghasilkan bilangan prima yang berbentuk $2kp + 1$ dengan k adalah bilangan asli dan bilangan prima ini kongruen dengan 1 atau 7 di modulo 8. Hal ini dikarenakan setiap faktor prima dari M_p pasti berbentuk $2kp + 1$ dan bersisa 1 atau 7 saat dibagi 8,

seperti yang sudah dibuktikan pada bagian sebelumnya.

2. Karena $M_p = 2^p - 1$, penentuan apakah M_p habis dibagi oleh sebuah bilangan prima q akan lebih cepat apabila dilakukan dengan menghitung semua bilangan yang berbentuk 2^{2^k} dalam modulo q dan kemudian mengalikannya sehingga menjadi sama dengan 2^p. Hal ini dapat didemonstrasikan dengan langkah-langkah berikut:
   a. Ubah p ke basis 2. Misalkan hasilnya adalah $p_2 = (a_1 a_2 \dots a_i)_2$ dengan $a_1, a_2, \dots, a_i \in \{0, 1\}$.
   b. Set nilai r sebagai 1.
   c. Set nilai j sebagai 1.
   d. Set nilai k sebagai nilai dari a_j.
      Jika $k = 1$, nilai r menjadi s^2 dengan s adalah sisa bagi saat $2r$ dibagi dengan q.
      Jika $k = 0$, nilai r menjadi s^2 dengan s adalah sisa bagi saat r dibagi dengan q.
   e. Jika $j < i$, ubah nilai j menjadi $j + 1$ dan ulangi ke langkah (d).
      Jika $j = i$, selesai.
   f. M_p akan habis dibagi oleh q apabila nilai r adalah 1.
3. Pengecekan faktor prima yang mungkin dari M_p akan memakan waktu yang sangat lama jika harus dicek sampai $\sqrt{M_p} \approx 2^{\frac{p}{2}}$, terutama untuk nilai $p > 20.000.000$. Oleh karena itu, Prime95 sudah menentukan batas atas dari bilangan prima yang akan dicek seperti yang ditunjukkan pada gambar 3.2. Bilangan Mersenne yang ternyata masih belum ditemukan faktor primanya akan ditentukan lagi dengan algoritma pemfaktoran Pollard $p - 1$ atau dengan pengetasan bilangan prima.

Exponents up to	Trial factor to
0	2^{65}
23,390,000	2^{66}
29,690,000	2^{67}
37,800,000	2^{68}
47,450,000	2^{69}
58,520,000	2^{70}
75,670,000	2^{71}
96,830,000	2^{72}
115,300,000	2^{73}
147,500,000	2^{74}
186,400,000	2^{75}
227,300,000	2^{76}
264,600,000	2^{77}
337,400,000	2^{78}
420,400,000	2^{79}
516,000,000	2^{80}

Gambar 3.2. Batas atas pengecekan bilangan prima pada Algoritma Pemfaktoran Trial.
https://www.mersenne.org/various/math.php

C. Algoritma Pemfaktoran Pollard $p - 1$

Algoritma Pemfaktoran Pollard $p - 1$ ini dapat digunakan untuk memfaktorkan sebuah bilangan komposit n dengan mencari sebuah bilangan q yang merupakan faktor dari n dan $q - 1$ adalah sebuah bilangan yang *powersmooth*.

Misalkan n adalah bilangan komposit yang ingin dicari pemfaktorannya. Algoritma ini diimplementasikan dengan langkah-langkah berikut:

1. Tentukan nilai batas atas B yang berupa bilangan asli yang cukup besar.
2. Definisikan
$$m = \prod_{\substack{p \text{ prima} \\ 1 \le p \le B}} p^{\lfloor \log_p B \rfloor}$$
(Bilangan ini tidak perlu dihitung secara langsung)
3. Pilih sebuah bilangan asli a sehingga $1 < a < n$.
4. Hitunglah $FPB(a, n)$.
   Jika FPB keduanya adalah 1, pergi ke langkah 5.
   Jika FPB keduanya bukan 1, a adalah faktor dari n. Selesai.
5. Hitunglah nilai $g = FPB(a^m - 1, n)$.
   Jika $g = 1$, pergi ke langkah 1 dan naikkan nilai B.
   Jika $g = n$, pergi ke langkah 3 dan ganti nilai a.
   Jika $g \neq 1$ dan $g \neq n$, g adalah faktor nontrivial dari n. Selesai.

Algoritma Pemfaktoran Pollard $p - 1$ ini didasari oleh Teorema Fermat Kecil yakni apabila p adalah bilangan prima dan a adalah bilangan yang relatif prima dengan p, maka $a^{p-1} - 1$ habis dibagi oleh p. Jika n adalah bilangan yang sudah dipastikan komposit dan p adalah salah satu faktor prima n yang ingin dicari, maka $FPB(a^{p-1}, n) > 1$.

Tentunya, nilai p pada awalnya tidak diketahui. Jika mencari nilai p secara berurutan dari bilangan prima terkecil ke bilangan prima yang lebih besar, maka akan memakan waktu yang kurang lebih sama dengan algoritma pemfaktoran trial. Hal ini dapat diatasi dengan memanfaatkan fakta bahwa $a^{c(p-1)}$ juga akan bersisa 1 saat dibagi oleh p. Sehingga dengan mengambil bilangan asli m yang habis dibagi oleh banyak bilangan berbentuk $p - 1$ dengan p adalah bilangan prima yang relatif kecil (lebih kecil dari sebuah bilangan asli B yang sudah ditentukan), bilangan $a^m - 1$ akan memiliki banyak faktor positif. Dari bilangan-bilangan prima kecil ini, diharapkan ada bilangan prima p sehingga $p - 1$ membagi m dan p membagi bilangan Mersenne yang dicari.

Algoritma Pemfaktoran Pollard $p - 1$ sangat sesuai digunakan pada bilangan Mersenne. Perhatikan bahwa setiap faktor prima dari $M_p = 2^p - 1$ pasti berbentuk $2kp + 1$. Sehingga, untuk setiap q faktor dari M_p, bilangan $q - 1$ setidaknya habis dibagi oleh p. Jika bilangan k ini merupakan hasil kali beberapa bilangan prima yang relatif kecil, maka $a^k - 1$ akan memiliki faktor persekutuan dengan M_p untuk a sebuah bilangan yang relatif prima dengan M_p.

Dalam algoritma yang digunakan oleh GIMPS, terdapat beberapa variasi yang dilakukan:

1. Nilai m diubah menjadi hasil kali semua $2Ep$ dengan E adalah hasil kali semua bilangan prima yang kurang dari B.
2. Nilai a ditetapkan menjadi 3. Hal ini akan memastikan nilai a selalu relatif prima dengan M_p karena $2^p - 1 \equiv (-1)^p - 1 \equiv -2 \pmod 3$ untuk p ganjil.

3. Terdapat fase algoritma Pollard $p-1$ lanjutan yang menambahkan batas atas kedua B_2. Batas atas lanjutan ini akan berguna apabila k, dengan $q = 2kp + 1$ faktor prima dari M_p, memiliki tepat satu faktor prima di antara B dan B_2 dan faktor prima lainnya lebih kecil dari B. Fase ini memerlukan memori yang cukup besar.

D. Algoritma Pemfaktoran Kurva Eliptik (ECM)

Algoritma Pemfaktoran Kurva Eliptik (*Eliptic-Curve Method*/ ECM) adalah algoritma pemfaktoran yang ditemukan oleh Hendrik Lentstra. Algoritma ECM dikhususkan untuk mendapatkan faktor-faktor kecil dari sebuah bilangan yang besar. Seperti namanya, algoritma ini menggunakan teori kurva eliptik.

Algoritma ini bekerja cukup mirip dengan algoritma Pollard $p-1$. Keduanya sama-sama memanfaatkan batas atas B dan bilangan $m = \prod_{\substack{p \text{ prima} \\ 1 \le p \le B}} p^{\lfloor \log_p B \rfloor}$. Perbedaannya adalah, alih-alih menghitung modulo, algoritma ECM memilih sebuah kurva eliptik secara acak, memilih titik di kurva ini secara acak, dan melakukan perhitungan dengan titik ini. Detailnya tidak akan ditulis di makalah ini karena sudah di luar topik.

E. Pengetesan Bilangan Prima Fermat

Pengetesan Bilangan Prima Fermat digunakan untuk mengeliminasi bilangan Mersenne yang jelas bukan bilangan prima. Pengetesan Bilangan Prima Fermat ini masih memungkinkan adanya bilangan komposit yang lulus tes ini. Bilangan yang lulus tes ini disebut sebagai *probable prime*.

Dalam pseudocode, pengetesan bilangan prima ini dapat ditulis sebagai:

```
Fermat(p, k)
   Globalcomposite ← false
   repeat k times:
         Localcomposite ← false
         Ambil bilangan asli b secara
         acak dengan 1 < b < p - 2
         if (b^p ≢ 1 (mod p)) then
               Localcomposite ← true
         if Localcomposite = true then
               Globalcomposite ← true
   if Globalcomposite = true then
         → COMPOSITE
   else
         → PROBABLE PRIME
```

Pengetesan Bilangan Prima Fermat ini memanfaatkan Teorema Fermat Kecil yang menyatakan bahwa $a^{p-1} \equiv 1 (\text{mod } p)$ untuk setiap bilangan prima p dan bilangan asli a yang relatif prima dengan p.

Permasalahan utama dari pengetesan prima ini adalah adanya bilangan pseudoprima Fermat dan bilangan Carmichael. Bilangan pseudoprima Fermat adalah bilangan komposit m sehingga terdapat sebuah bilangan asli a yang memenuhi $a^{m-1} \equiv 1 \pmod m$. Bilangan Carmichael adalah bilangan komposit m yang untuk setiap bilangan asli a yang relatif prima dengan m, berlaku $a^{m-1} \equiv 1 \pmod m$.

Untungnya, bilangan pseudoprima Fermat dan bilangan Carmichael lumayan jarang ditemui. Banyak bilangan pseudoprima Fermat yang lebih kecil dari 1.000.000 adalah 245. Bilangan Carmichael jauh lebih jarang. Banyak bilangan Carmichael yang lebih kecil dari 25×10^9 adalah 2163.

F. Pengetesan Bilangan Prima Lucas-Lehmer

Pengetesan Bilangan Prima Lucas-Lehmer dapat digunakan untuk menentukan apakah sebuah bilangan Mersenne $M_p = 2^p - 1$ dengan p merupakan bilangan prima adalah sebuah bilangan prima atau tidak.

Definisikan barisan bilangan asli $\{s_i\}_{i \ge 0}$ dengan aturan

$$s_i = \begin{cases} 4, & \text{jika } i = 0 \\ s_{i-1}^2 - 2, & \text{jika } i > 0. \end{cases}$$

Bilangan Mersenne $M_p = 2^p - 1$, dengan p merupakan bilangan prima, adalah sebuah bilangan prima Mersenne jika dan hanya jika s_{p-2} habis dibagi oleh M_p.

Dalam pseudocode, pengetesan bilangan prima ini dapat ditulis sebagai:

```
Lucas-Lehmer(p)
     s ← 4
     M ← 2^p − 1
     repeat p − 2 times:
           s ← ((s × s) - 2) mod M
     if s = 0 then
           → PRIME
     else
           → COMPOSITE
```

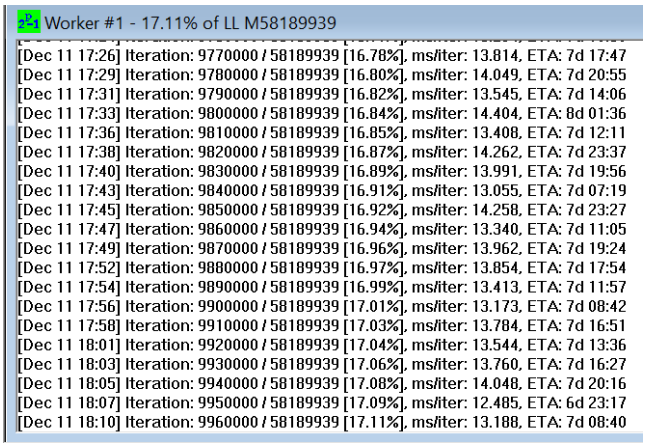
Dalam menentukan sifat sebuah bilangan Mersenne $M_p = 2^p - 1$, perlu diadakan tepat p buah iterasi. Setiap iterasi, program menghitung nilai dari $s^2 - 2$ dalam modulo M_p. Permasalahan utama di sini adalah bagaimana cara menghitung $s^2 - 4$ modulo M_p ini sedangkan M_p sendiri, dan kemungkinan s, memiliki jutaan digit. GIMPS sendiri menggunakan metode Fast Fourier Transform (FFT) yang menggunakan basis bilangan *float*. Implementasi FFT pada algoritma ini ditulis secara spesifik menggunakan bahasa pemrograman assemble agar bekerja lebih cepat pada cache arsitektur Intel Pentium. Karena menggunakan basis bilangan *float* inilah, implementasi algoritma Lucas-Lehmer perlu diverifikasi oleh pengguna yang lain.

Pengetesan Bilangan Prima Lucas-Lehmer inilah yang merupakan algoritma pengetesan prima utama yang digunakan sukarelawan GIMPS. Waktu yang dibutuhkan untuk menentukan apakah sebuah bilangan Mersenne prima atau tidak memang masih memakan waktu yang cukup lama seperti ditunjukkan pada gambar 3.3. Untuk melakukan iterasi Lucas-Lehmer sebanyak 50-60 juta kali, dibutuhkan waktu 2 sampai 57 hari. Hal ini akan berlipat ganda apabila iterasi yang dilakukan mencapai 100 juta kali; waktu yang dibutuhkan dapat mencapai ratusan hari.

Member Name	Computer Name	M_N	Type	UTC Time Received	Days	GHz-days
-Anonymous-		60320189	C-LL	2020-12-11 09:59	12.0	134.0030
-Anonymous-		60320633	C-LL	2020-12-11 09:59	12.0	134.0040
satar2	VC62B	54930019	C-LL	2020-12-11 09:49	20.4	108.5249
carlds	baremetal1	59327729	C-LL	2020-12-11 09:48	25.7	128.7700
TAMUC-ComputerScience	751-124300	54907427	C-LL	2020-12-11 09:44	23.2	108.4803
choppy07		56791291	C-LL	2020-12-11 09:32	57.0	120.3660
M. Farrokhi D. G.	farrokhi	55038271	C-LL	2020-12-11 09:28	6.0	108.7388
tdulcet	Colab_CPU	59372557	C-LL	2020-12-11 09:06	23.9	131.8978
-Anonymous-	hp	104626321	C-LL	2020-12-11 08:57	46.0	416.5799
Kendall		101293217	C-LL	2020-12-11 08:55	14.2	386.3211
mandelbrot31415	DESKTOP	55054561	C-LL	2020-12-11 08:54	5.3	108.7710
tottenhoff	Skyscraper-004	54139637	C-LL	2020-12-11 08:47	2.3	106.9634
Oytun	work	103977119	C-LL	2020-12-11 08:45	263.8	426.4506
-Anonymous-	Ofi	104062447	C-LL	2020-12-11 08:45	27.7	426.8006
sloops	Fal	55011871	C-LL	2020-12-11 08:33	9.7	108.6866
Mark L.		54791521	C-LL	2020-12-11 08:32	38.3	108.2513
P_B_MCL	HSWL_PBM	54332413	C-LL	2020-12-11 08:28	14.9	107.3442
David Barina	nbbarina	56538901	C-LL	2020-12-11 08:22	5.0	119.8311
GrunwalderGIMP	NewQC	54117211	C-LL	2020-12-11 08:07	6.0	106.9191
Chun sung soo	eunkueng_Kang	104816233	C-LL	2020-12-11 08:01	338.6	417.3360
Across-The-Stars	Ja	59693957	C-LL	2020-12-11 07:59	19.8	132.6118
TerminaTore	Lappy	59113601	C-LL	2020-12-11 07:43	5.8	131.3225
-Anonymous-		55028891	C-LL	2020-12-11 07:38	7.3	108.7203

Gambar 3.3. Hasil pengetesan keprimaan Lucas-Lehmer dari beberapa sukarelawan GIMPS (Kolom M_N menunjukkan bilangan Mersenne $2^N - 1$ yang diuji)
https://www.mersenne.org/report_recent_results/ diakses 11 Desember 2020, 17:50

Penulis juga ikut mencoba mengikuti proyek GIMPS ini. Penulis mendapatkan tugas untuk memverifikasi bahwa bilangan $M_{58189939} = 2^{58189939} - 1$ adalah sebuah bilangan komposit dengan menggunakan algoritma Lucas-Lehmer. Perangkat komputer yang Penulis miliki adalah sebuah laptop Asus Zenbook UM431-DA dengan processor AMD Ryzen 5 3500U yang dilengkapi Radeon Vega Mobile Gfx 2.10 GHz. Hasilnya, laptop Penulis dapat menjalankan iterasi dengan kecepatan 12-14 ms per iterasi dalam keadaan laptop membuka aplikasi lain dan 10-11 ms per iterasi dalam keadaan laptop senggang. Dengan asumsi laptop menjalankan perhitungan 24 jam per hari, diperkirakan waktu yang dibutuhkan untuk menyelesaikan perhitungan adalah 7-10 hari (Penulis telah menjalankan program ini sebelumnya selama 1,5 hari) . Lengkapnya dapat dilihat pada gambar 3.4.



Gambar 3.4. Percobaan pemverifikasian kekompositan bilangan Mersenne dengan algoritma Lucas-Lehmer
Dokumentasi pribadi

IV. KESIMPULAN

Dalam mencari bilangan prima Mersenne yang memiliki eksponen sangat besar (> 40.000) diperlukan berbagai pengetahuan mengenai algoritma pengetesan bilangan prima dan algoritma pemfaktoran yang cepat dan efisien. Pendistribusian pekerjaan pemfaktoran, pengetesan keprimaan, dan pemverifikasian hasil akan mempercepat kemajuan proyek kolosal pencarian bilangan prima Mersenne berekponen besar ini. Waktu yang dibutuhkan untuk mencari bilangan prima Mersenne atau memfaktorkan bilangan komposit Mersenne berekponen puluh-jutaan ini, bahkan dengan ilmu pengetahuan paling mutakhir, masih memakan waktu mingguan, bulanan, bahkan sampai tahunan. Hal ini dapat dijadikan acuan capaian ilmu pengetahuan kolektif umat manusia dalam bidang teori bilangan dan ilmu komputasi.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa beserta orangtua dan Bapak serta Ibu dosen yang telah selalu mendukung dan membimbing Penulis untuk menyelesaikan makalah ini.

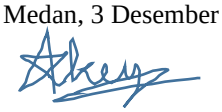
DAFTAR PUSTAKA

[1] T. Andreescu, *Number Theory: Structures, Examples, and Problems*. Basel: Birkhäuser, 2009, ch. 9.
[2] A. Charest, "Pollard's p-1 and Lenstra's factoring algorithms (Unpublished work style)," unpublished, 2005.
[3] D. M. Bressoud, *Factorization and Primality Testing*. New York: Springer, 1989.
[4] https://www.mersenne.org diakses berkali-kali pada 3-11 Desember 2020
[5] R. Crandall, C. Pomerance, *Prime Numbers: A Computational Perspective*. New York: Springer-Verlag, 2001

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Medan, 3 Desember 2020



Akeyla Pradia Naufal - 13519178