Frobenius Problem Beserta Solusinya

Muhammad Gilang Ramadhan 13520137
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13520137@std.stei.itb.ac.id

***Abstrak*—Frobenius Problem dapat dikatakan sebagai sebuah masalah besar yang ada di dalam dunia matematika, khususnya pada bidang teori bilangan. Banyak matematikawan dunia yang tidak berhasil memberikan solusi pada permasalahan ini secara eksplisit. Namun, bukan berarti masalah ini tidak dapat diselesaikan sama sekali, melainkan bisa diselesaikan secara implisit melalui pendekatan-pendekatan tertentu. Dalam makalah ini penulis akan menyajika beberapa solusi yang mungkin bisa ditawarkan untuk Frobenius Problem. Pada makalah ini, penulis memberikan bukti secara matematis pada setiap langkah yang diambil agar langkah yang diambil tersebut menjadi logis.**

***Kata kunci*—Frobenius Problem, Solusi, Teori Bilangan.**

I. PENDAHULUAN

Ferdinand George Frobenius adalah seorang matematikawan yang lahir pada tanggal 26 Oktober 1849 di Berlin, Jerman. Dalam kiprahnya di dunia matematika, Beliau dikenal lewat karyanya pada Teori Fungsi Eliptik, Persamaan Differensial, Teori Grup, dan Teori bilangan. Pada pokok bahasan kali ini, penulis akan membahas karya yang telah dibuatnya pada bidang teori bilangan, yaitu Frobenius Number.

Menurut definisinya, Frobenius Number adalah bilangan bulat positif terbesar yang tidak bisa dinyatakan sebagai kombinasi linear bilangan bulat nonnegatif tertentu yang saling relatif prima. Menurut sejarahnya, Frobenius Number ini muncul ketika Ia terpikir sebuah masalah yang dikenal dengan nama Frobenius Problem. Frobenius Problem ini awalnya dibahas oleh Frobenius pada akhir tahun 1800-an dalam perkulihaannya, tetapi pada saat itu Ia tidak pernah mempublikasikannya.

Kemudian, pada tahun 1882 Sylvester, menemukan sebuah masalah sekaligus memberikan solusi kepada masalah yang berhubungan dengan Frobenius Problem tersebut. Jika ditinjau masalah yang ditemukan oleh Sylvester tersebut sebenarnya sama dengan Frobenius Problem. Namun, pada kasus Sylvester tersebut hanya dilibatkan 2 bilangan bulat saja. Setelah itu, muncul sebuah masalah yang berkenaan dengan masalah Sylvester tersebut, yaitu Chicken McNugget Problem yang melahirkan Chicken McNugget Theorem. Selain itu, sebenarnya terdapat berbagai masalah lain yang berkaitan dengan Frobenius Number, contohnya Postage Stamp Problem dan Frobenius Coin Problems. Oleh karena itu, pada makalah ini penulis tertarik dalam membahas solusi yang bisa ditawarkan untuk memecahkan masalah tersebut.

Gambar 1.1. Ferdinand George Frobenius

(Sumber : https://peoplepill.com/people/ferdinand-georg-frobenius)

II. TEORI TERKAIT

2.1 Linear Diophantine Equation

Menurut definisinya, Diophantine Equation (Persamaan Diophantine) merupakan sebuah persamaan polinom yang semua variabelnya merupakan elemen bilangan bulat. Sedangkan Linear Diophantine dapat diartikan sebagai polinom berorde satu yang solusi dari variabelnya merupakan elemen bilangan bulat atau secara matematis dapat dinyatakan sebagai berikut.

$$a_1x_1 + a_2x_2 + \cdots + a_kx_k = b$$

Dengan $a_1, a_2, \ldots, a_k, b \in \mathbb{Z}$ dan $a_i \neq 0$ untuk setiap $i = 1,2, \ldots, k$.

2.2 Induksi Matematika

Induksi Matematika adalah cara paling mudah yang digunakan untuk membuktikan suatu identitas atau formula tanpa menggunakan secara langsung. Menurut penggunaannya ada beberapa jenis induksi, antara lain induksi lemah, induksi kuat, dan induksi tak beraturan. Namun, induksi yang digunakan oleh penulis dalam pokok bahasan kali ini adalah induksi lemah. Menurut prinsipnya, induksi dapat diibaratkan dengan efek domino; jika domino pertama dijatuhkan, akan memberikan efek pada domino yang lain untuk jatuh. Berikut disajikan langkah-langkah dalam menggunakan induksi.

1. Pertama, buktikan bahwa basis dari induksi benar. Basis disini adalah indeks pertama dari identitas atau formula yang bersangkutan.
2. Merancang hipotesis yang berkatian dengan formula atau identitas yang bersangkutan. Dalam merancang hipotesis tersebut berarti dugaan tersebut dianggap benar untuk $n = k$.
3. Langkah terakhir adalah membuktikan hipotesis yang telah dirancang tersebut dengan membuktikan pernyataan tersebut benar untuk $n = k + 1$.

2.3 Kombinatorika

Dalam kombinatorika, terdapat dua fundamental yang merupakan pokok dari semua teori yang ada, yaitu kombinasi dan permutasi.

1. Kombinasi
   Kombinasi dapat diartikan sebagai cara untuk menentukan banyaknya kemungkinan dalam pengambilan k buah objek dari n buah objek dengan $k \leq n$ tanpa memperhatikan susunannya. Secara matermatis, kombinasi dapat dinyatakan sebagai berikut.

$$C_k^n = \frac{n!}{k!\,(n-k)!}$$

2. Permutasi
   Sama halnya dengan kombinasi, permutasi merupakan banyaknya cara dalam pengambilan k buah objek dari n buah objek dengan $k \leq n$. Namun, pada permutasi susunan pada pengambilan diperhatikan, dengan demikian apabila terdapat dua susunan yang berbeda pada suatu pengambilan, maka kedua susunan itu dianggap sebagai dua konfigurasi yang berbeda. Secara matematis, permutasi dapat dinyatakan sebagai berikut.

$$P_k^n = \frac{n!}{(n-k)!}$$

2.4 Binomial Newton

Binomial Newton adalah metode aljabar yang dipakai untuk menjabarkan suatu ekspresi binomial. Pada konsep Binomial Newton, koefisien dari setiap suku dari hasil penjabaran ekspresi binomialnya dapat dinyatakan sebagai kombinasi dari banyaknya suku dan urutan suku. Secara matematis, Binomial Newton dapat dinyatakan sebagai berikut.

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} \cdot a^{n-i} \cdot b^i$$

Dengan $k, i \in \mathbb{N}$.

Bukti :
Akan dibutkikan dengan menggunakan induksi matematika,

1. Untuk n = 0, maka diperoleh :

$$(a+b)^0 = \binom{0}{0} \cdot a^{0-0} \cdot b^0$$

$$\Leftrightarrow 1 = 1 \text{ (jelas benar)}$$

2. Andaikan untuk $n = k$ benar, maka diperoleh :

$$(a+b)^k = \sum_{i=0}^{k} \binom{k}{i} \cdot a^{k-i} \cdot b^i$$

   Haruslah bernilai benar (hipotesis).

3. Akan dibuktikan untuk $n = k + 1$, ekspresi tersebut bernilai benar. Substitusi $n = k + 1$, maka diperoleh :

$$(a+b)^{k+1} = \sum_{i=0}^{k+1} \binom{k+1}{i} \cdot a^{k+1-i} \cdot b^i$$

   Perhatikan bahwa

$$(a+b)^{k+1} = (a+b)^k(a+b)$$

$$\Leftrightarrow (a+b)^{k+1} = \sum_{i=0}^{k} \binom{k}{i} \cdot a^{k-i} \cdot b^i \cdot (a+b)$$

   Misalkan

$$\sum_{i=0}^{k} \binom{k}{i} \cdot a^{k-i} \cdot b^i \cdot (a+b) = T$$

   Maka diperoleh :

$$T = a^{k+1} + \cdots + \left[\binom{k}{k-1} + \binom{k}{k}\right] ab^k + b^{k+1}$$

$$\Leftrightarrow T = \binom{k+1}{0} a^{k+1} + \cdots + \binom{k+1}{k} ab^k + \binom{k+1}{k+1} b^{k+1}$$

$$\Leftrightarrow T = \sum_{i=0}^{k+1} \binom{k+1}{i} \cdot a^{k+1-i} \cdot b^i$$

   Yang mengakibatkan

$$(a+b)^{k+1} = \sum_{i=0}^{k+1} \binom{k+1}{i} \cdot a^{k+1-i} \cdot b^i. \square$$

2.5 Division Algorithm

Jika terdapat $a, b \in \mathbb{Z}$ dan $b > 0$ maka terdapat bilangan bulat unik $q, r \in \mathbb{Z}$ sedemikian sehingga terdapat dua kondisi, yaitu :

$$a = bq + r \text{ dan } 0 \leq r \leq b$$

Pada kondisi ini, variabel a bisa dinyatakan sebagai bilangan yang dibagi, b sebagai pembagi, q sebagai hasil bagi, dan r sebagai sisa pembagian.

Bukti :
Pertama-tama , akan dibuktikan eksistensi dari q dan r.
Di soal diberikan $b > 0$ dan untuk setiap a, definisikan

$$q = \left\lfloor \frac{a}{b} \right\rfloor$$

$$r = a - bq$$

Sehingga dapat diperoleh $a = bq + r$. Mengingat yang akan dibuktikan adalah $0 \leq r < b$. Maka dapat dideduksikan Lemma sebagai berikut.

Lemma 2.5.1 :
Untuk setiap $x \epsilon \mathbb{Z}$, dapat diperoleh :

$$x - 1 < \lfloor x \rfloor \leq x$$

Bukti :
Misalkan $n = \lfloor x \rfloor$, maka dapat diperoleh $\lfloor x \rfloor \leq x$ dan $x < n + 1$, yang mengakibatkan $x - 1 < n = \lfloor x \rfloor$. ☐

Dari Lemma 2.5.1 diperoleh :

$$\Leftrightarrow \frac{a}{b} - 1 < \left\lfloor \frac{a}{b} \right\rfloor \leq \frac{a}{b}$$

$$\Leftrightarrow b - a > -b \left\lfloor \frac{a}{b} \right\rfloor \geq -a$$

$$\Leftrightarrow b > a - bq \geq 0$$

Karena $r = a – bq$, maka haruslah $0 \leq r < b$.
Untuk membuktikan bahwa q dan r unik dapat dicari.
Asumsikan bahwa

$$a = bq_1 + r_1 \text{ dan } 0 \leq r_1 < b$$

dan

$$a = bq_2 + r_2 \text{ dan } 0 \leq r_2 < b$$

Maka cukup buktikan bahwa $r_1 = r_2$ dan $q_1 = q_2$.
Andaikan bahwa $r_1 \neq r_2$, WLOG $r_2 > r_1$. Apabila $bq_1 + r_1$ dikurang $bq_2 + r_2$ maka diperoleh :

$$b(q_1 - q_2) + (r_1 - r_2) = 0$$

$$\Leftrightarrow b(q_1 - q_2) = (r_2 - r_1)$$

Yang berakibat $b|(r_2 - r_1)$, maka haruslah $b \leq r_2 - r_1$.
Padahal menurut asumsi $b \leq r_1 < r_2 < b$, $\Leftrightarrow b > (r_2 - r_1)$ (kontradiksi). Maka haruslah $r_2 = r_1$, akibatnya $b(q_1 - q_2) = r_2 - r_1 = 0$. Karena $b > 0 \Rightarrow q_1 = q_2$. ☐

2.6 Algoritma Euclid

Algoritma Euclid merupakan algoritma yang dapat dipakai dalam menentukan faktor persekutuan terbesar (Greatest Common Divisor) dari dua bilangan. Algoritma ini menggunakan sifat-sifat dari sisa pembagian yang dilakukan secara berpola. Adapun langkah-langkah dari Algoritma Euclid untuk menentukan gcd (a, b) adalah sebagai berikut.

$$a = bq_1 + r_1, \qquad 0 \leq r_1 < b$$

$$b = r_1q_2 + r_2, \qquad 0 \leq r_2 < r_1$$

$$r_1 = r_2q_3 + r_3, \qquad 0 \leq r_3 < r_2$$

…………….

$$r_{n-2} = r_{n-1}q_n + r_n, \qquad 0 \leq r_n < r_{n-1}$$

$$r_{n-1} = r_nq_{n+1}$$

Maka $\gcd(a, b) = r_n$, dengan $a, b, r_i, q_j \,\epsilon\, \mathbb{Z}$ untuk setiap $i = 1,2, \dots, n$ dan $j = 1,2,3, \dots, n + 1$

2.7 Fermat's Little Theorem

Jika a adalah bilangan bulat, p adalah bilangan prima dan a tidak membagi p, maka diperoleh :

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

Bukti:
Akan dibuktikan identitas Fermat's Little Theorem itu benar dengan menggunakan induksi. Untuk membuktikannya a dapat dijadikan parameter indeksnya. Berikut adalah langkah-langkah pembuktiannya.

1. Buktikan basisnya terlebih dahulu, subtitusi $a = 1$ diperoleh :
$$1^p \equiv 1 \text{ (mod } p)$$
Karena $1^p = 1$ untuk p berapapun, akibatnya diperoleh :
$$1 \equiv 1 \text{ (mod } p)$$
Jelas bahwa ekspresi tersebut benar untuk $a = 1$.
2. Andaikan untuk $a = k$ (hipotesis)
$$k^p \equiv k \text{ (mod } p)$$
Asumsikan pernyataan tersebut benar, dan cukup buktikan bahwa untuk $a = k + 1$ benar.
3. Akan dibuktikan untuk $a = k + 1$ benar, subtitustikan $a = k + 1$, maka diperoleh :
$$(k + 1)^p \equiv k + 1 \text{ (mod } p)$$
Tinjau $(k + 1)^p$ dengan Binomial Newton
$$(k + 1)^p = \sum_{i=0}^{p} \binom{p}{i} k^{p-i}$$
Perhatikan bahwa untuk setiap $0 < k < p,\ k \in \mathbb{N}$
$$\binom{p}{k} = \frac{p!}{(p - k)!\,k!}$$
Karena untuk setiap $k, (p - k)$ tidak membagi p dan k juga tidak membagi p, akibatnya diperoleh :
$$\tbinom{p}{k} \equiv 0 \text{ (mod } p)$$
$$\Rightarrow (k + 1)^p \equiv \tbinom{p}{0} k^p + 1 \text{ (mod } p)$$
Dari subtitusi hipostesis sebelumnya ke persamaan di atas, maka diperoleh :
$$\Leftrightarrow (k + 1)^p \equiv k + 1 \text{ (mod } p)$$
Persamaan tersebut merupakan persaaman yang ingin dibuktikan. ☐

2.8 Teorema Euclid Kedua

Adapun Teorema Euclid Kedua yang berbunyi “Ada takhingga banyaknya bilangan prima”.

Bukti Euclid :

Misalkan diberikan bilangan prima konsekutif 2,3,5, … , P. Definisikan juga N sebagai bilangan prima sedemikian sehingga

$$N = (2 \times 3 \times 5 \times \dots \times P) + 1$$

dengan demikian menurut Algoritma Euclid diperoleh:

$$\gcd(N, 2 \times 3 \times 5 \times \dots \times \mathrm{P}) = 1$$

akibatnya karena $N > P$, maka pasti terdapat bilangan prima lain yang lebih besar dari P, sehingga dapat deduksikan bahwa ada takhingga banyaknya bilangan prima adalah benar. $\square$

2.9 Lemma Bezout

Untuk suatu bilangan bulat nonnegatif a dan b, maka terdapat bilangan bulat x dan y sedemikian sehingga

$$ax + by = \gcd(a, b)$$

Bukti :

Tinjau Himpunan

$$P = \{ n \in \mathbb{Z}^{+} | n = a_1x_1 + a_2x_2 \in \mathbb{Z} \}$$

Dengan $P \neq \emptyset$. Karena semua elemen dari P adalah positif, Sehingga pastilah terdapat $d \in P$ dengan d adalah elemn minimum dari himpunan P. Sehingga diperoleh :

$$d = a_1x_1 + a_2x_2$$

Jika $n > d$ dan $n \in S$ maka diperoleh:

$$n = a_1u_1 + a_2u_2$$

Tetapi dengan *Division Algorithm*

$$n = qd + r \Rightarrow r = n - qd$$

$$\Leftrightarrow n = a_1(u_1 - qx_1) + a_2(u_2 - qx_2)$$

$$r \in P$$

Akibatnya diperoleh $0 \leq r < d$, padahal asumsi diawal d adalah elemen terkecil di P (kontradiksi). Akibatnya haruslah $r = 0$, Sehingga untuk setiap $i \in \{1,2\}$, diperoleh:

$$a_i = a_i \cdot 1 + \sum_{k=1; k \neq i}^{2} (a_k \cdot 0) \Rightarrow \{a_1, a_2\} \subset P$$

maka dapat dideduksikan $d \mid n$ yang berakibat $d \mid a_i$

Sehingga diperoleh:

$$d \mid \gcd(a_1, a_2) \mid a_i$$

$$\gcd(a_1, a_2) \mid \sum_{i=1}^{2} a_iu_i = d$$

Dengan demikian diperoleh $d = \gcd(a_1, a_2)$. $\square$

2.10 Algoritma Greedy

Algoritma Greedy merupakan suatu metode yang bisa digunakan untuk menyelesaikan berbagai permasalahan yang berkaitan dengan optimasi.

Berikut disajikan elemen-elemen dari Algoritma Greedy, yaitu :

1. Himpunan Kandidat (C).
2. Himpunan Solusi (S).
3. Fungsi Seleksi. (*Selection Function*).
4. Fungsi Kelayakan (*Feasible*).
5. Fungsi Obyektif.

Dengan demkian dapat disimpulkan bahwa :

Algoritma Greedy adalah suatu algoritma yang melibatkan pencarian himpunan bagian (subhimpunan) (S) dari himpunan kandidat (C) yang pada prinsipnya harus memenuhi beberapa syarat, yaitu menyatakan suatu solusi dan S dapat dioptimasi dengan fungsi obyektif.

Berikut disajikan Notasi Algoritmik dari skema umum Algoritma Greedy.

```
{ Skema Umum Algoritma Greedy }
function Greedy_Algorithm(input C: Himpunan kandidat) -> Himpunan
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy
  Input : himpunan kandidat C,
  Output : himpunan solusi yang bertipe himpunan_kandidat,
  mengembalikan -1 jika tidak ada solusi}
KAMUS LOKAL
    x: kandidat
    S: himpunan_kandidat
ALGORITMA
S <- {}  { inisialisasi S dengan kosong }
while (not SOLUSI(S)) and (C != {}) do
    x <- SELEKSI(C)         { Pilih sebuah kandidat dari C}
    C <- C - {x}            { Elemen himpunan kandidat berkurang satu }
    if LAYAK(concat(S,{x})) then
        S <- concat(S,{x})
{SOLUSI(S) or C = {} }
if SOLUSI(S) then
    -> S
else
    -> -1 { Jika tidak ada solusi }
```

Gambar 2.10.1 Skema Umum Algoritma Greedy

(Sumber : Dokumen Pribadi)

III. FROBENIUS PROBLEM

3.1 Masalah

Diberikan $x_1, x_2, x_3, \dots, x_n \in \mathbb{Z}$ dengan $\gcd(x_1, x_2, x_3, \dots, x_n) = 1$, hitunglah bilangan bulat positif terbesar yang tidak bisa direpresentasikan sebagai kombinasi linear bilangan bulat nonnegatif a_i terhadap $x_i \geq 0$ dengan $i \,\epsilon\, \{1,2,3, \dots, n\}$ atau dalam notasi lain masalah tersebut ekuivalen dengan mencari b terbesar sedemikian sehisngga tidak ada solusi x_i pada persamaan :

$$\sum_{i=1}^{n} a_ix_i = b$$

$$\gcd(x_1, x_2, \dots, x_n) = 1$$

Dengan $x_i, b \geq 1, a_i \geq 0, a_i, x_i, b \,\epsilon\, \mathbb{Z}$

3.2 Solusi Matematika

3.2.1 Untuk $n = 1$

Tidak ada bilangan bulat yang memenuhi
Bukti :
Substitusi $n = 1$, maka diperoleh :

$$a_1x_1 = b$$

Apabila diambil b adalah bilangan prima dan x_1 tidak membagi b maka jelas bahwa $a_1x_1 \neq b$. Perhatikan juga bahwa menurut Teorema Euclid kedua bahwa banyaknya bilangan prima ada takhingga banyaknya, akibatnya banyak solusi b yang memenuhi ada takhingga banyaknya juga, sehingga tidak ada bilangan bulat b yang memenuhi. ☐

3.2.2 Untuk $n = 2$

Subsitutsi $n = 2$, maka diperoleh :

$$a_1x_1 + a_2x_2 = b$$

Sehingga bilangan bulat positif terbesar b yang memenuhi adalah $x_1x_2 - x_1 - x_2$.

3.2.2.1 Bukti Pertama :

Lemma 3.2.1 :
Misalkan $A_b \subset \mathbb{Z} \times \mathbb{Z}$ adalah himpunan solusi (a_1, a_2) dari $a_1x_1 + a_2x_2 = b$. Maka diperoleh $A_b = \{(a_1 + kx_2, a_2 - kx_1)\}$, untuk setiap $(a_1, a_2) \in A_b$ dan $k \in \mathbb{Z}$.
Bukti :
Menurut Lemma Bezout, terdapat bilangan bulat x', y' sedemikian sehingga

$$a_1'x_1 + a_2'x_2 = 1$$

$$\Leftrightarrow ba_1'x_1 + ba_2'x_2 = b$$

Karena A_b tidak kosong, maka mudah dicek bahwa $(ba_1' + kx_2, ba_2' - kx_1) \in A_b$ untuk setiap $k \in \mathbb{Z}$.

Sekarang akan dibuktikan bahwa tidak ada kemungkinan lain yang memenuhi. Andaikan (a_1', a_2') dan (a_1'', a_2'') adalah solusi dari $a_1x_1 + a_2x_2 = b$. Maka $a_1'x_1 + a_2'x_2 = a_1''x_1 + a_2''x_2$ yang berakibat $x_1(a_1' - a_1'') = x_2(a_2'' - a_2')$. Dikarenakan x_1 dan x_2 adalah relatif prima serta $x_1 \mid x_2(a_2'' - a_2')$, maka $x_1 \mid a_2'' - a_2' \Rightarrow a_2'' \equiv a_2' \pmod{x_1}$.

Dengan cara yang sama akan diperoleh $a_1'' \equiv a_1' \pmod{x_2}$. Misalkan k_1, k_2 adalah bilangan bulat yang memenuhi $a_1'' - a_1' = k_1x_2$ dan $a_2'' - a_2' = k_2x_1$. Maka didapat $x_1(-k_1x_2) = x_2(k_2x_1)$ yang berakibat $k_1 = -k_2$. Sehingga didapat hasil yang diinginkan. ☐

Lemma 3.2.2 :
Untuk setiap bilangan bulat N, terdapat solusi unik $a_N, b_N) \in \mathbb{Z} \times \{0,1, \dots, m-1\}$ sedemikian sehingga $a_Nm + b_Nn = N$.
Bukti :
Dengan Division Algorithm, dapat dideduksikan bahwa hanya satu $k \in \mathbb{Z}$ sedemikian sehingga $0 \leq y - km \leq m - 1$. ☐

Lemma 3.2.3 :
N pasti dapat dinyatakan dalam bentuk

$$\sum_{i=1}^{n} a_ix_i$$

jika dan hanya jika $a_N \geq 0$, untuk setiap $x_i \geq 1, a_i \geq 0, a_i, x_i \in \mathbb{Z}$.

Bukti:
Jika $a_N \geq 0$, maka bisa diambil $(a, b) = (a_N, b_N)$ sedemikian sehingga N dapat dinyatakan dalam ekspresi tersebut. Jika $a_N < 0$ maka $a_N + kn < 0$ dicapai ketika $k \leq 0$ dan $b_N - km < 0$ dicapai ketika $k > 0$. Perhatikan juga bahwa paling sedikit satu di antara $a_N + kn$ atau $b_N - km$ akan bernilai negatif untuk semua $k \in \mathbb{Z}$. Dengan demikian, didapat N tidak dapat dinyatakan dalam ekspresi tersebut. ☐

Sehingga, dari Lemma 3.2.1, Lemma 3.2.2, dan Lemma 3.2.3, diperoleh $xm + yn$ dengan $x < 0$ dan $0 \leq y \leq m - 1$, dapat dinyatakan sebagai kombinasi linear suatu bilangan bulat nonnegatif. Dengan demikian maksimum dari $xm + yn$ dicapai ketika $x = -1$ dan $y = m - 1$, sehingga didapat $xm + yn = (-1)m + (m-1)n = mn - m - n$. ☐

(Sumber : https://artofproblemsolving.com/)

3.2.2.2 Bukti Kedua :

Misalkan $S = \{1,2,3, \dots, p-1\}$. Apabila masing-masing elemen dari S dikalikan dengan a atau bisa dituliskan dengan $a \cdot S$, maka dengan Fermat's Little Theorem, diperoleh :

$$S \equiv \{1 \times a, 2 \times a, \dots, (p-1) \times a\} \pmod{p}$$

Karena jelas bahwa tidak ada $(i \times a)$ untuk setiap $1 \leq i \leq p - 1$ yang habis dibagi p, akibatnya semua elemen di $a \cdot S$ pasti berbeda. Tinjau $a_i \equiv a_j \pmod{p}$ untuk setiap $i \neq j$. Perhatikan juga bahwa $\gcd(a, p) = 1$, maka dengan mereduksinya dapat dideduksikan $i \equiv j \pmod{p}$ (kontradiksi).

Karena m dan n adalah relatif prima, maka dengan mengalikan residu dari m dengan n hanya akan mengubah residunya. Tinjau bahwa masing-masing permutasi dari residu tersebut merupakan bilangan yang dapat dinyatakan sebagai kombinasi linear suatu bilangan bulat nonnegatif (Dengan menggunakan definisi dari bukti pertama), karena pada ekspresi $am + bn$, a bernilai 0 dan b adalah original residu, maka cukup buktikan lemma berikut.

Lemma 3.2.4 :
Untuk setiap bilangan bulat nonnegatif c dan $c < m$, cn adalah bilangan terkecil yang dapat dinyatakan sebagai kombinasi linear dari suatu bilangan bulat nonnegatif yang kongruen cn modulo m.
Bukti :
Perhatikan bahwa setiap bilangan yang kurang dari cn dan kongruen cn mod m bisa direpresentasikan dengan ekspresi $cn - dm$, dengan d adalah bilangan bulat positif. Apabila bilangan tersebut dapat dinyatakan sebagai kombinasi linear dari suatu bilangan bulat nonnegatif, maka didapat $cn - dm = am + bn$ untuk suatu bilangan bulat nonnegatif a, b. Sehingga, bisa ditulis ulang ke bentuk ekspresi $(a + d)m = (c - b)n$, yang

berakibat bahwa $(a+d)$ adalah kelipatan dari n (karena $\gcd(m,n)=1$). Dari argumen tersebut diperoleh $(a+d)=gn$ untuk suatu $g \in \mathbb{Z}$ dan $g>0$ dan bila disubtitusikan m ke $(a+d)=gn$, maka diperoleh $gmn=(c-b)n$. Apabila ditinjau $c<m, (c-b)n<mn$, dan $gmn<mn$, $\Leftrightarrow$ $g<1$. Padahal $g \in \mathbb{Z}$ dan $g \geq 1$ (kontradiksi). Akibatnya cn adalah bilangan terkecil yang dapat dinyatakan dengan kombinasi linear suatu bilangan bulat nonnegatif yang kongruen cn modulo m. ☐

Dari Lemma 3.2.4, dapat dideduksikan bahwa cn dapat dinyatakan dengan kombinasi linear suatu bilangan bulat nonnegatif, karena untuk setiap bilangan lebih besar dari cn dan kongruen dengan cn modulo m pasti dapat dinyatakan sebagai kombinasi linear dari bilangan bulat nonnegatif juga (karena bilangan tersebut bisa dinyatakan dengan ekspresi $am+bn$ dengan $b=c$). Dari Lemma tersebut juga dapat dideduksikan $cn-m$ adalah bilangan terbesar yang kongruen cn mod m tidak dapat dinyatakan sebagai kombinasi linear dari suatu bilangan bulat nonnegatif, maka diperoleh :

$$c \leq m-1$$

$$\Leftrightarrow cn-m \leq mn-m-n$$

Dengan demikian, diperoleh $mn-m-n$ adalah bilangan bulat terbesar dari $cn-m$. Karena bilangan tersebut lebih besar dari cn, akibatnya semua bilangan yang lebih besar dari bilangan tersebut dan kongruen cn mod m pasti tidak dapat dinyatakan sebagai sebagai kombinasi linear suatu bilangan bulat nonnegatif. Dengan demikian untuk semua bilangan yang kongruen dengan cn dan lebih besar dari $mn-m-n$ dapat dinyatakan sebagai kombinasi linear suatu bilangan bulat nonnegatif.

Sekarang apabila semua fakta tersebut disatukan, maka dapat dideduksikan bahwa untuk setiap m dan n relatif prima, $mn-m-n$ adalah bilangan bulat terbesar yang tidak bisa direpresentasikan kedalam ekspresi $am+bn$ untuk setiap a,b bilangan bulat nonnegatif. ☐

(Sumber : https://artofproblemsolving.com/)

3.2.2.3 Bukti Ketiga :

Andaikan $a_1x_1+a_2x_2$ bisa dinyatakan dalam bentuk $x_1x_2-x_1-x_2$. Maka untuk setiap bilangan bulat nonnegative x_1, x_2, apabila ditinjau modulo x_1, diperoleh :

$$a_2x_2 \equiv -x_2 \pmod{x_1}$$

$$\Leftrightarrow a_2 \equiv -1 \pmod{x_1}$$

Dengan cara yang sama juga diperoleh :

$$a_1x_1 \equiv -x_1 \pmod{x_2}$$

$$\Leftrightarrow a_1 \equiv -1 \pmod{x_2}$$

Karena a_1 dan a_2 adalah bilangan bulat nonnegatif maka diperoleh :

$$x_1(x_2-1)+x_2(x_1-1) \leq a_1x_1+a_2x_2$$

Karena

$$2x_1x_2-x_1-x_2 > x_1x_2-x_1-x_2$$

Maka haruslah

$$a_1x_1+a_2x_2 > x_1x_2-x_1-x_2$$

Karena diawal asumsinya $a_1x_1+a_2x_2$ bisa dinyatakan dalam bentuk $x_1x_2-x_1-x_2$ (kontradiksi).

Akibatnya pastilah $b \geq x_1x_2-x_1-x_2$.

Sekarang, tinjau $b > x_1x_2-x_1-x_2$, Dengan, Lemma Bezout, terdapat a_1' dan a_2' sedemikian sehingga $a_1'x_1+a_2'x_2=1$, $\Leftrightarrow$ $ba_1'x_1+ba_2'x_2=b$ $\Rightarrow$ $a_1''=ba_1{}'$ dan $a_2''=ba_2{}'$ $\Rightarrow$ $x_1a_1''+x_2a_2''=b$. Sehingga solusi dari $a_1x_1+a_2x_2=b$ adalah $a_1=a_1''+tx_2$, $a_2=a_2''-tx_1$ dengan $t \in \mathbb{Z}$.

Dengan Division Algorithm, bisa dipilih t sedemikian sehingga $0 \leq a_1 \leq x_2-1$ $\Rightarrow a_1x_1+a_2x_2=b>x_1x_2-x_1-x_2 \Rightarrow$ $x_2(a_2+1)>x_1(x_2-1-a_1)$. Mengingat $x_1,x_2>0$ dan $x_2-1-a_1 \geq 0$, akibatnya untuk setiap $a_1,a_2 \geq 0$ terdapat solusi a_1 dan a_2 sedemikian sehingga $a_1x_1+a_2x_2=b$.

Sekaligus membuktikan $b=x_1x_2-x_1-x_2$. ☐

(Sumber : https://brilliant.org/)

3.2.3 Untuk $n \geq 3$

Formula b nya tidak dapat dinyatakan secara eksplisit, tetapi tetap dapat dicari dengan Teorema Johnson berikut.

Misalkan $g(x_1,x_2,\dots,x_n)=b$, $\gcd(x_2,x_3,\dots,x_n)=d$, d $\neq$ 1 serta $x_j=dx_j'$ untuk setiap $j>1$, maka dapat dideduksikan relasi rekursif sebagai berikut.

$$g(x_1,x_2,\dots,x_n)=d \cdot g(x_1,x_2',\dots,x_n')+x_1(d-1)$$

Bukti :

Untuk membuktikan Teorema Johnson tersebut akan digunakan Lemma sebagai berikut.

Lemma 3.2.5 :
Misalkan $\gcd(a_1,a_2,\dots,a_k)=1$, dan untuk setiap $1 \leq i \leq a_1-1$, misalkan juga m_i sebagai bilangan bulat postif *N* terkecil yang kongkuren *i* modulo a_1 sedemikian sehingga $a_1x_1+a_2x_2+\cdots+a_kx_k=N$ mempunyai solusi bilangan bulat nonnegatif, maka

$$g(a_1,a_2,\dots,a_k)=\max_{1 \leq i \leq a_1-1} m_i-a_1$$

Bukti :
Dari definisi m_i, dapat dikatakan bahwa m_i-a_1 tidak bisa dinyatakan oleh bilangan bulat nonnegatif $a_1,a_2,\dots,a_k$, untuk setiap *i*, dengan $1 \leq i \leq a_1$. Perhatikan juga bahwa, untuk setiap *N* dan $N>m_i-a_1$ serta $N \equiv i \pmod{a_1}$, akibatnya haruslah m_i terkecil dan m_i dapat dinyatakan oleh bilangan bulat nonnegatif $a_1,a_2,a_3,\dots,a_k$. ☐

Maka dari Lemma 3.2.5, untuk setiap $1 \leq i \leq x_1-1$, misalkan

m_i dan m_i' dapat dinyatakan sebagai bilangan bulat positif terkecil yang kongruen i modulo x_1 sedemikian sehingga dapat dinyatakan sebagai kombinasi linear dari masing-masing $x_1, x_2, x_3, \dots, x_n$ dan $x_1, x_2', x_3', \dots, x_n'$. Karena, masing-masing m_i dan m_i' juga harus bisa dinyatakan ke dalam kombinasi linear dari masing-masing bilangan bulat nonnegatif $x_2, x_3, \dots, x_n$ dan $x_2', x_3', \dots, x_n'$, mengakibatkan $m_i = dm_i'$, untuk setiap $1 \le j \le x_1 - 1$. Sehingga, dengan menggunakan Lemma 3.2.5, diperoleh:

$$g(x_1, x_2, \dots, x_n) = \max_{1 \le j \le x_1 - 1} m_j - x_1$$

$$\Leftrightarrow g(x_1, x_2, \dots, x_n) = d \cdot \left(\max_{1 \le j \le x_1 - 1} m_j' - x_1 \right) + x_1(d - 1)$$

$$\Leftrightarrow g(x_1, x_2, \dots, x_n) = d \cdot g(x_1, x_2', \dots, x_n') + x_1(d - 1). \square$$

Solusi rekursif tersebut merupakan solusi umum secara implisit yang bisa dipakai untuk $n \ge 1$.

(Sumber : https://web.iitd.ac.in/~atripath/publications/FP.pdf)

3.3 Solusi Pemrograman

Untuk mencari solusi dari Frobenius Number tersebut dapat menggunakan Algoritma Greeedy. Caranya dengan merekontruksi masalah kecil dari subhimpunan bilangan $x_1, x_2, \dots, x_n$ dengan suatu koefisien vektor $(a_1, a_2, \dots, a_n)$ sedemikian sehingga :

$$\sum_{i=1}^{n} a_i x_i = a \cdot x = b$$

Dengan $a \cdot x$ merupakan dot product, untuk suatu $b \in \mathbb{Z}$. Perhatikan bahwa ekspresi diatas dapat dicapai ketika $a_i = 0$, untuk setiap $i = 1, \dots, n - 1$ serta memenuhi ekspresi

$$a_n = \left\lfloor \frac{b}{x_n} \right\rfloor$$

Apabila didefinisikan $t \equiv b - a \cdot x$, maka algoritma selesai ketika sudah mencapai $\Delta = 0$, jika belum mencapai $\Delta = 0$, maka lakukan operasi berikut.

$$a_i = \left\lfloor \frac{b}{x_i} \right\rfloor$$

Untuk setiap $i = i - 1, i - 2, \dots, 1$ sampai diperoleh $\Delta = 0$.

Berikut disajikan Kode dalam bentuk Notasi Algoritmik yang merupakan Implementasi dari Algoritma Greedy tersebut.

3.3.1 Algoritma Alternatif Pertama

```
{ ALGORITMA 1 }
function Frobenius_Number(input x[n]: array, input f: integer) -> integer
{ x[n] sebagai array yang merupakan x_1,x_2,...,x_n dari frobenius problem,
sedangkan f sebagai batas upper bound dari frobenius number }
KAMUS LOKAL
    E[f]: array
    t, FrobNumber: integer
ALGORITMA
    FrobNumber <- 0
    i traversal [1...f]
        t <- 0
        j traversal [1...n]
            t <- t + E[i-x[j]]
        E[i] <- t
        if (t = 0) then
            FrobNumber <- i
    if (FrobNumber > 0) then
        -> FrobNumber
```

Gambar 3.3.1 Notasi Algoritmik Algoritma 1

(Sumber : Dokumen Pribadi)

Keterangan : Kompleksitas Algoritma $O(n^2)$

3.3.2 Algoritma Alternatif Kedua

```
{ ALGORITMA 2 }
function Frobenius_Number(input x[n]: array, input f: integer) -> integer
{ x[n] sebagai array yang merupakan x_1,x_2,...,x_n dari frobenius problem,
sedangkan f sebagai batas upper bound dari frobenius number }
KAMUS LOKAL
    E[f]: array
    t, FrobNumber: integer
ALGORITMA
    FrobNumber <- 0
    i traversal [1...f]
        t <- 0
        j traversal [1...n]
            t <- t + E[i-x[j]]
        if (t = 0) then
            E[i] <- 0
            FrobNumber <- i
        else
            E[i] <- 1
    if (FrobNumber > 0) then
        -> FrobNumber
```

Gambar 3.3.2 Notasi Algoritmik Algoritma 2

(Sumber : Dokumen Pribadi)

Keterangan : Kompleksitas Algoritma $O(n^2)$

3.3.3 Algoritma Alternatif Ketiga

```
{ ALGORITMA 3 }
function Frobenius_Number(input x[n]: array, input f: integer) -> integer
{ x[n] sebagai array yang merupakan x_1,x_2,...,x_n dari frobenius problem,
sedangkan f sebagai batas upper bound dari frobenius number }
KAMUS LOKAL
    E[f]: array
    l, t, u, FrobNumber: integer
    max(x): integer { Maksimum dari x_1,x_2,...,x_n }
ALGORITMA
    l <- max(x)
    FrobNumber <- 0
    u <- l + 1
    i traversal [1...f]
        t <- 0
        j traversal [1...n]
            t <- t + E[u-x[j]]
        if (t = 0) then
            E[i] <- 0
            FrobNumber <- i
        else
            E[i] <- 1
        k traversal [2...u]
            E[k-1] <- E[k]
    if (FrobNumber > 0) then
        -> FrobNumber
```

Gambar 3.3.3 Notasi Algoritmik Algoritma 3

(Sumber : Dokumen Pribadi)

Keterangan : Kompleksitas Algoritma $O(n^2)$

IV. KESIMPULAN

Dari semua solusi yang dipaparkan oleh penulis, Frobenius Problem dapat diselesaikan secara eksplisit hanya pada kasus $n = 1$ dan $n = 2$ saja. Sedangkan secara umum, solusi yang bisa ditawarkan oleh penulis adalah solusi implisitnya saja. Pada solusi implisit yang disajikan oleh penulis pada makalah ini pun tidak bisa mencakup semua kasus yang ada, melainkan hanya beberapa kasus saja. Hal tersebut dikarenakan terdapat kasus kosong pada relasi tersebut. Adapun pada makalah ini, penulis juga menggunakan Algoritma Greedy sebagai solusi lain dalam menyelesaikan Frobenius Problem ini. Dengan mengetahui Ide yang terdapat pada permasalahan ini, penulis berharap ada solusi lain yang lebih optimal selain solusi yang dipaparkan oleh penulis pada makalah ini di masa depan.

V. UCAPAN TERIMA KASIH

Pertama-tama, Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat-Nya penulis dapat menyelesaikan makalah ini dengan baik, Penulis juga berterimakasih kepada orang tua, serta teman-teman yang selalu memberikan semangat dan dukungan kepada penulis sehingga makalah ini dapat terselesaikan. Penulis juga tak lupa berterimakasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., Ibu Harlili, M.Sc, dan Ibu Dr. Nur Ulfa Maulidevi S.T., M.Sc. selaku dosen mata kuliah matematika diskrit yang telah memberikan banyak ilmu dan motivasi dalam kegiatan perkuliahan. Terakhir, penulis memohon maaf apabila dalam penulisan makalah ini terdapat kesalahan baik disengaja maupun tidak disengaja. Penulis berharap makalah ini dapat bermanfaat bagi banyak orang.

REFERENSI

[1] Jean-Pierre Serre, “Georg Frobenius” [online] Tersedia : https://www.britannica.com/biography/Georg-Frobenius, diakses tanggal 6 Desember 2021.

[2] www.artofproblemsolving.com, diakses tanggal 6 Desember 2021.

[3] www.brilliant.org, diakses tanggal 6 Desember 2021.

[4] Amitabha Tripathi, “ON A LINEAR DIOPHANTINE PROBLEM OF FROBENIUS” [online]. Tersedia : https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.487.3114&rep=rep1&type=pdf, diakses tanggal 6 Desember 2021.

[5] Naoki Sato, “Number Theory” [online]. Tersedia : https://artofproblemsolving.com/articles/files/SatoNT.pdf, diakses tanggal 7 Desember 2021.

[6] Weisstein, Eric W. "Frobenius Number." [online] Tersedia : https://mathworld.wolfram.com/FrobeniusNumber.html, diakses tanggal 7 Desember 2021.

[7] Rinaldi Munir, “Algoritma Greedy” [online]. Tersedia: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-(2018).pdf, diakses tanggal 11 Desember 2021.

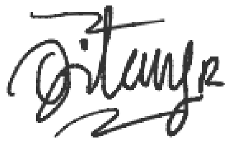
[8] Weisstein, Eric W, “Greedy Algorithm” [online]. Tersedia : https://mathworld.wolfram.com/GreedyAlgorithm.html, diakses tanggal 11 Desember 2021.

[9] Charles Sauerbier, “Computing a Frobenius Coin Problem decision in $O(n^2)$” [online]. Tersedia: https://arxiv.org/ftp/arxiv/papers/1001/1001.0961.pdf, diakses tanggal 11 Desember 2021.

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.

Bandung, 12 Desember 2021



Muhammad Gilang Ramadhan 13520137