

# Penerapan Graf Sebagai Solusi Alternatif untuk Memecahkan Masalah *Chicken McNugget Theorem*

Rabbi Fajar Mayoza, 13516081  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia  
13516081@std.stei.itb.ac.id

**Abstrak**—*Chicken McNugget Theorem* merupakan sebuah teorema yang dicetuskan oleh seorang matematikawan, Henri Picciotto, pada tahun 1980-an sebagai petunjuk atas pencarian nilai *Frobenius Number*. *Frobenius Number* merupakan sebuah solusi dari sebuah persamaan *diophantine*. Dalam mencari nilai *Frobenius Number*, hingga saat ini matematikawan di seluruh dunia masih berusaha menemukan berbagai cara dan algoritma yang paling mangkus. Pemecahan masalah *Frobenius Number* dapat mencakup banyak pokok bahasan dalam Matematika Diskrit, seperti Teori Bilangan Bulat, Induksi Matematika, Graf, dan lain-lain. Oleh karena itu, pada makalah ini penulis membatasi ulasan dengan memfokuskan kajian pada penerapan graf sebagai solusi alternatif untuk memecahkan masalah *Chicken McNugget Theorem*. Metode ini ditemukan oleh Albert Nijenhuis bersama 3 peneliti lainnya dari berbagai negara bagian di Amerika Serikat di awal abad 21 ini.

**Kata Kunci**—*Frobenius Number*, *Chicken McNuggets Theorem*, *Nijenhuis graph*, pohon merentang minimum.

## I. PENDAHULUAN

*Frobenius problem* merupakan teka-teki klasik untuk menemukan sebuah bilangan yang disebut *Frobenius Number*, yaitu bilangan terbesar yang tidak mungkin menjadi solusi dari persamaan *diophantine Frobenius Equation*. Permasalahan *Frobenius Number* ini sering dijumpai dalam kehidupan sehari-hari, contohnya menentukan apakah prangko senilai 700 dan 1000 rupiah dapat digunakan untuk membayar biaya pos sebesar 1500 rupiah. Tentunya permasalahan seperti contoh tersebut merupakan kasus sederhana sehingga dapat dengan mudah dipecahkan dengan logika manusia. Namun untuk bilangan sembarang yang sangat besar atau sangat banyak dibutuhkan algoritma yang mangkus agar permasalahan *Frobenius Number* dapat diselesaikan dengan komputer. Berbagai macam teori dan algoritma telah dikemukakan oleh berbagai matematikawan di dunia, salah satunya yang paling menarik menurut penulis adalah *Chicken McNugget Theorem* dan *Nijenhuis Graph*.

### A. *Frobenius Equation*

*Frobenius Equation* adalah sebuah pemodelan matematika yang berawal dari permasalahan mencari pecahan terbesar yang tidak dapat dihasilkan dari kombinasi sejumlah koin dengan nilai berbeda dalam bentuk persamaan *diophantine*:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b ;$$

dengan  $a_n, x_n$ , dan  $b$  merupakan *integer* bukan negatif. *Frobenius Number* merupakan nilai  $b$  terbesar sedemikian sehingga *Frobenius Equation* tidak memiliki solusi. Artinya untuk setiap bilangan lebih dari  $b$ , maka solusi akan selalu ada. Nama *Frobenius Equation* diambil dari penemu persamaan tersebut, Ferdinand Georg Frobenius, seorang matematikawan berkebangsaan Jerman, pada abad ke-19. Sampai saat ini, belum ada algoritma baku untuk mencari nilai *Frobenius Number*. Akan tetapi mayoritas matematikawan modern sepakat untuk menggunakan rumus yang dikemukakan oleh seorang matematikawan berkebangsaan Inggris, James Joseph Sylvester (1884).

$$g(a_1, a_2) = (a_1 - 1)(a_2 - 1) - 1 \\ = a_1 a_2 - (a_1 + a_2).$$

Kekurangan dari rumus ini yaitu hanya menangani kasus untuk  $n = 2$ .

### B. *Chicken McNugget Theorem*

Teorema ini merupakan kasus khusus dari *Frobenius Problem*. Teorema ini diperkenalkan pertama kali oleh seorang matematikawan, Henri Picciotto, pada tahun 1980-an. Teorema ini bernama demikian karena Picciotto menemukannya saat ia bersama anaknya sedang makan bersama di restoran cepat saji McDonald's. Mereka berdua memesan *Chicken McNugget*. Pada masa itu di Inggris, *Chicken McNugget* dijual dalam tiga ukuran kotak yang berbeda: kotak berukuran kecil berisi 6 potong, kotak berukuran sedang berisi 9 potong, dan kotak berukuran besar berisi 20 potong. Saat makan tiba-tiba terlintas pemikiran di benak Picciotto, bagaimana jika ia ingin memesan *Chicken McNugget* dalam jumlah tertentu untuk dibagikan kepada sanak keluarga atau teman-teman saat mengadakan pesta. Menurut Picciotto, pramusaji pasti akan menolak pesanan seperti itu karena menurut mereka tidak boleh menjual *nugget* yang jumlahnya bukan kelipatan dari ukuran kotak.

Menariknya ketiga ukuran kotak tersebut ( $\{6,9,20\}$ ) adalah relatif prima. Picciotto kemudian mengambil selembar serbet dan mencoret-coretinya untuk menentukan *Frobenius Number* dari permasalahan *Chicken McNugget* ini. Ternyata dari coret-coretan tersebut Picciotto menemukan *Frobenius Number*-nya dan lahirlah *Chicken McNugget Theorem*.

**Chicken McNugget Theorem:**

All integers are McNugget numbers except 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 22, 23, 25, 28, 31, 34, 37, and 43. They are called 'non-McNugget numbers'. Thus the largest non-McNugget number is 43. [Picciotto, 1994]

Ini berarti siapapun bisa memesan *Chicken McNugget* sebanyak berapapun asalkan lebih dari 43 potong. Sayangnya teorema tersebut mengalami sedikit perubahan sejak McDonald's mengeluarkan *Chicken McNugget* ukuran baru yang berisi 4 potong *nugget* sehingga 4,8, dan 10 sekarang termasuk *McNugget numbers* dan *non-McNugget number* terbesarnya menjadi 11 (*Frobenius Number* dari himpunan  $\{4,6,9,20\}$  adalah 11). Di beberapa negara bahkan muncul varian ukuran lain. Walaupun demikian konsep dari teorema ini tetap sama, yaitu menentukan *Frobenius Number*.

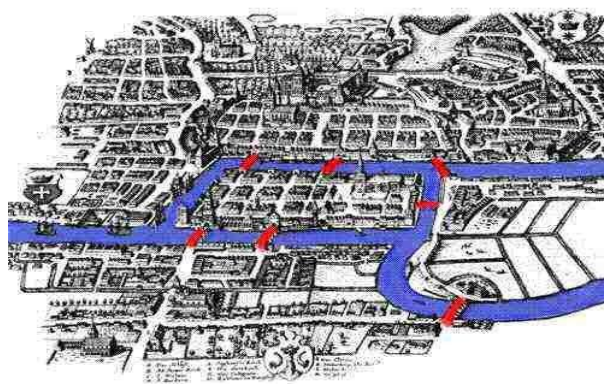


Gambar 1.1. Chicken McNuggets Box berukuran 20 potong  
Sumber : wikipedia.org

**II. LANDASAN TEORI**

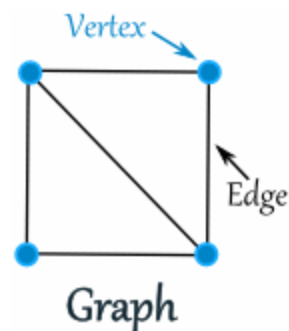
**A. Graf**

Konsep dari teori graf pertama kali diperkenalkan pada tahun 1736 untuk menyelesaikan permasalahan Jembatan Königsberg. Leonhard Euler, seorang matematikawan Swiss, mempelajari permasalahan ini dan kemudian berhasil membangun suatu solusi yang kemudian melahirkan konsep dari Eulerian Graph. Sampai saat ini, Euler dianggap sebagai peletak dasar teori graf dan diberi gelar Bapak Teori Graf.



Gambar 2.1. Jembatan Königsberg  
Sumber : maa.org

Graf  $G$  didefinisikan sebagai pasangan himpunan  $(V,E)$  dimana  $V$  merupakan himpunan tak-kosong dari simpul (*vertex*) dan  $E$  merupakan himpunan sisi (*edge*) yang menghubungkan simpul-simpul pada  $V$ .

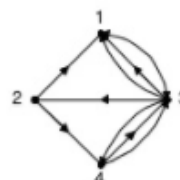


Gambar 2.2. Simpul dan Sisi pada Graf  
Sumber: mathisfun.com

**B. Jenis Graf**

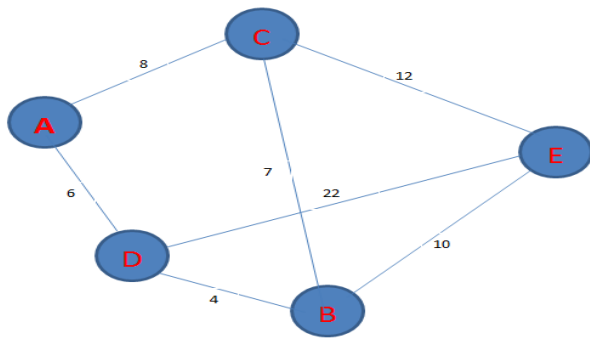
Jenis-jenis graf yang akan digunakan dalam pembahasan topik makalah ini adalah graf ganda-berarah.

- Graf Ganda  
Berbeda dari graf sederhana yang hanya memiliki sisi tunggal, graf ganda adalah graf yang mempunyai sisi ganda.
- Graf Berarah  
Graf yang sisinya diberi orientasi arah (disebut juga "busur") dinamakan graf berarah.



Gambar 2.3. Graf ganda-berarah  
Sumber : <http://informatika.stei.itb.ac.id/~rinaldi.munir>

### C. Terminologi dalam Teori Graf



Gambar 2.4. Contoh Graf  
Sumber : [generasi-nias.blogspot.co.id/2014/07](http://generasi-nias.blogspot.co.id/2014/07)

- Bertetangga  
Dua buah simpul dikatakan bertetangga jika ada sebuah sisi yang menghubungkan mereka. Pada gambar 2.4, simpul A dan C dikatakan bertetangga.
- Bersisian  
Sebuah simpul dan sebuah sisi dikatakan bersisian jika salah satu ujung dari sisi adalah simpul yang dimaksud. Pada gambar 2.4, simpul A bersisian dengan sisi (A,D) dan (A,C).
- Derajat  
Derajat suatu simpul adalah banyaknya sisi yang bersisian dengan simpul tersebut, dengan kalang dihitung dua kali. Pada gambar 2.4, derajat dari simpul C adalah 3.
- Lintasan  
Lintasan adalah barisan sisi yang menghubungkan satu simpul dengan simpul lain.
- Sirkuit  
Pada suatu graf, sirkuit adalah lintasan yang bermula dan berakhir pada simpul yang sama.
- Terhubung  
Dua buah simpul dikatakan terhubung jika ada lintasan yang menghubungkan keduanya. Pada gambar 2.4, simpul A terhubung dengan simpul B. Pada graf berarah terbagi menjadi terhubung kuat dan terhubung lemah. Contohnya yaitu pada gambar 2.3 simpul 1 dan 3 terhubung kuat karena terdapat lintasan dari 1 ke 3 dan dari 3 ke 1. Sedangkan simpul 1 dan 2 terhubung lemah karena hanya terdapat lintasan dari 2 ke 1.
- Upagraf  
Misal  $G = (V,E)$  adalah sebuah graf, maka  $G_1 = (V_1,E_1)$  adalah upagraf dari  $G$  jika  $V_1$  dan  $E_1$  masing-masing merupakan himpunan bagian dari  $V$  dan  $E$ .
- Himpunan bebas  
Himpunan bebas (*independent set*) adalah serangkaian simpul pada graf sehingga tidak ada dua simpul yang bertetangga. Pada gambar 2.4,  $\{C,D\}$  adalah salah satu himpunan bebas yang

memungkinkan.

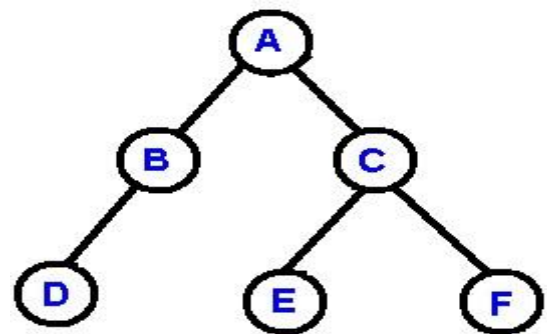
- Graf Berbobot  
Graf berbobot adalah graf yang sisi-sisinya diberi label yang berisi sebuah harga. Gambar 2.4 merupakan contoh graf berbobot.

### D. Pohon

Pohon merupakan graf terhubung yang tidak mengandung sirkuit. Dengan kata lain, pohon adalah graf khusus. Istilah “pohon” pertama kali diperkenalkan oleh matematikawan Inggris, Arthur Cayley, pada tahun 1857, sebagai solusi ketika ia menghitung jumlah senyawa kimia. Karena definisi pohon diacu dari teori graf, maka pohon juga dapat mempunyai hanya sebuah simpul tanpa sisi. Graf  $G = (V,E)$  dapat didefinisikan sebagai pohon apabila memenuhi syarat berikut.

- Setiap pasang simpul di dalam  $G$  terhubung dalam lintasan tunggal
- $G$  memiliki sisi sejumlah simpul  $- 1$ .
- $G$  tidak mengandung sirkuit dan penambahan satu sisi pada graf akan menciptakan hanya satu sirkuit.

### E. Terminologi Pohon Berakar



Gambar 2.5. Contoh Pohon Berakar  
Sumber: [thebutterflyboy.files.wordpress.com](http://thebutterflyboy.files.wordpress.com)

Karena topik bahasan makalah ini mengaplikasikan graf berarah, maka dalam bentuk pohon graf tersebut menjadi pohon berakar. Pohon berakar adalah pohon yang simpulnya diperlakukan sebagai akar dan sisi-sisinya berarah menjauhi akar. Sebagai konvensi, arah sisi dalam pohon tidak perlu digambar karena pada pohon berakar setiap simpul hanya dapat dicapai melalui akarnya dari atas ke bawah.

Berikut adalah beberapa terminologi dari Pohon Berakar.

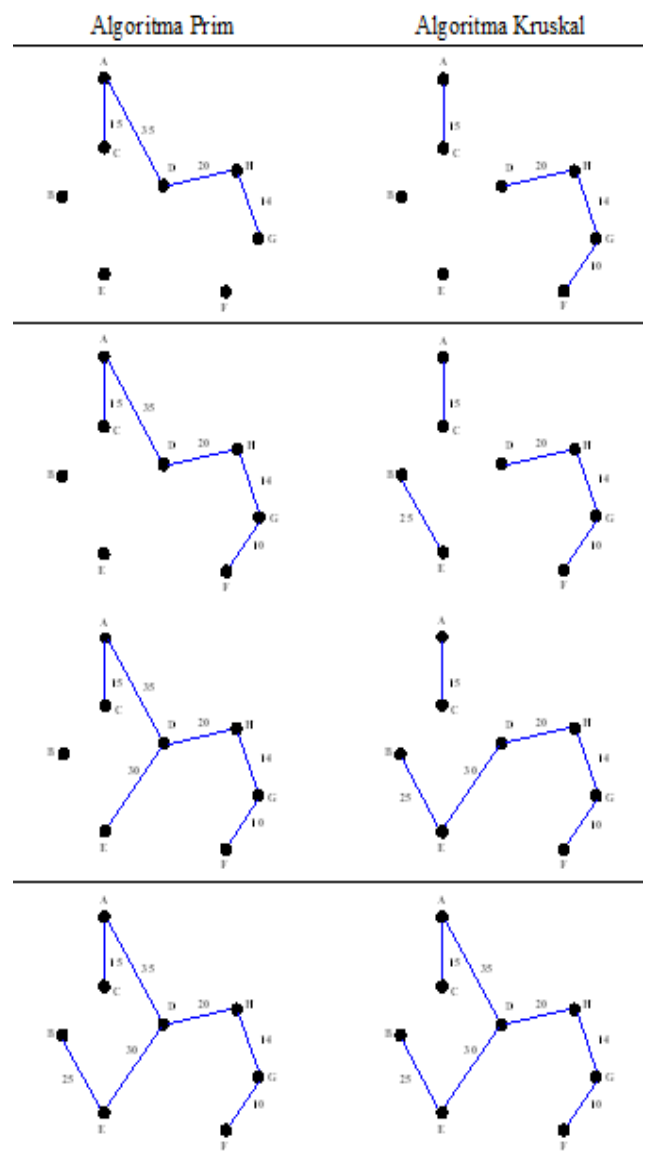
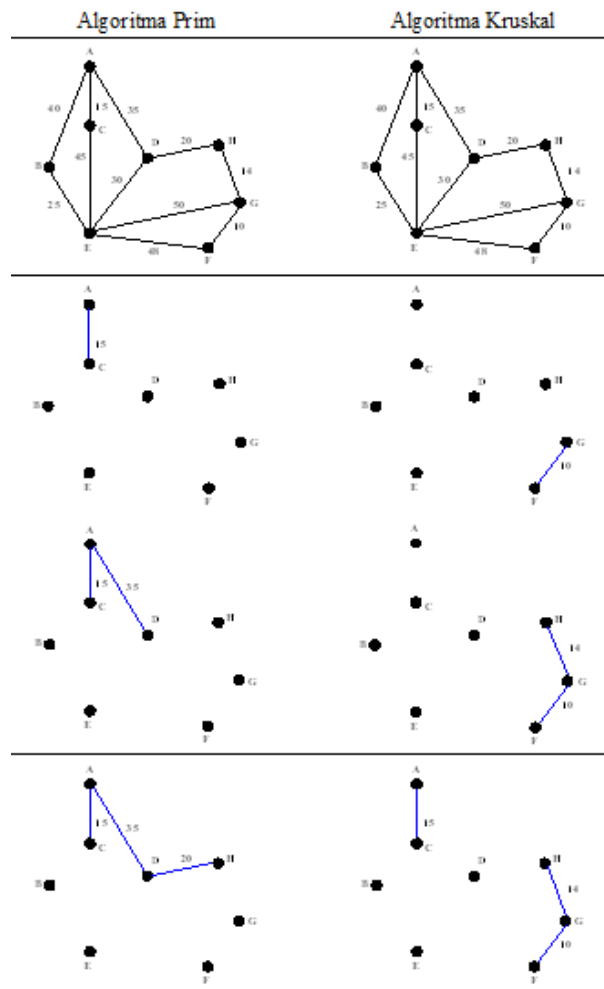
- *Child* dan *Parent*  
Pada gambar 2.5, simpul B merupakan *child* dan simpul A sebagai *parent*-nya karena simpul B terhubung dari akarnya yaitu simpul A. Begitu juga dengan simpul D yang merupakan *child* dari simpul B dan simpul B yang merupakan *parent* dari simpul D.
- Lintasan  
Lintasan merupakan runtunan simpul dan sisi sehingga antara *parent* dengan *child*-nya saling

terhubung. Pada gambar 2.5, panjang lintasan dari A ke D adalah 2.

- Derajat  
Derajat dari suatu simpul adalah jumlah upapohon atau *child* dari simpul tersebut. Pada gambar 2.5, derajat simpul A adalah 2 dan derajat simpul B adalah 1.
- Daun  
Daun adalah simpul berderajat 0. Pada gambar 2.5, simpul {D,E,F} adalah daun.
- Tinggi atau Kedalaman  
Tinggi atau kedalaman adalah panjang maksimum lintasan dari akar ke daun. Pada gambar 2.5, tinggi dari pohon adalah 3.

F. *Pohon Merentang Minimum*

Misalkan  $G = (V,E)$  adalah graf terhubung yang bukan pohon.  $G$  kemudian diubah menjadi pohon  $T$  dengan memutuskan sirkuit-sirkuit di  $G$ . Bila semua sirkuit di  $G$  sudah hilang, maka pohon  $T$  yang terbentuk dinamakan pohon merentang. Pohon merentang bisa dikatakan sebagai upagraf dari  $G$  yang berbentuk pohon. Sedangkan pohon merentang minimum adalah pohon merentang dari graf berbobot yang bobotnya paling kecil. Terdapat dua algoritma membangun pohon merentang minimum. Yang pertama adalah Algoritma Prim dan yang kedua adalah Algoritma Kruskal.



Gambar 2.6. Perbedaan Algoritma Prim dan Kruskal dalam Membangun Pohon Merentang Minimum  
Sumber : [dwipuspita53.wordpress.com/2014/01/30/perbedaan-algoritma-prim-dan-kruskal/](http://dwipuspita53.wordpress.com/2014/01/30/perbedaan-algoritma-prim-dan-kruskal/)

III. THE FROBENIUS CIRCULANT GRAPH MODEL

*The Frobenius Circulant Graph Model* atau *Nijenhuis Graph* merupakan metode yang ditemukan oleh matematikawan Albert Nijenhuis dari Washington bersama dengan tiga peneliti lainnya yaitu Dale Beihoffer dari Minnesota, Jemimah Henry dari Wisconsin, dan Stan Wagon dari Minnesota pada tahun 2005. Tujuan dari penelitian mereka adalah mencari algoritma tercepat untuk menyelesaikan *Frobenius Equation*. Bisa dikatakan



keunggulan dari metode ini adalah kecepatannya dalam menyelesaikan *Frobenius Equation*.

Diberikan sebuah *Frobenius Equation*,

$$x_1 a_1 + x_2 a_2 + \dots + x_n a_n = M;$$

dengan  $a_i$  dan  $M$  adalah *integer* positif. *Frobenius number*  $f(A)$ , dengan  $A = \{a_i\}$  adalah  $M$  terbesar sedemikian sehingga persamaan ini tidak memiliki solusi.  $M$  merupakan *Frobenius Number* jika dan hanya jika  $\gcd(a_1, a_2, \dots, a_n) = 1$ . Misal diberikan basis  $A = \{a_1, a_2, \dots, a_n\}$  himpunan *integer* positif, maka  $M$  *representable* jika terdapat himpunan himpunan *integer* positif  $X = \{x_i\}$  sedemikian sehingga

$$\sum_{i=1}^n a_i x_i = M.$$

(1)

Untuk selanjutnya, kita akan menggunakan istilah *representable* dan *non representable* untuk  $M$ .

Untuk memodelkan *Frobenius Equation* ini ke dalam bentuk graf, maka kita harus memproses  $M$  sebagai mod dari  $a_1$  sehingga (1) diubah ke dalam bentuk

$$\sum_{i>1} a_i x_i \equiv M \pmod{a_1}.$$

(2)

untuk basis  $A = (a_1, a_2, \dots, a_n)$ , didefinisikan graf  $G(A)$  sebagai graf berbobot-berarah dengan nomor simpul  $\{0, \dots, a_1 - 1\}$  maka jumlah keseluruhan simpulnya adalah  $a_1$ . Kemudian kita misalkan  $u$  dan  $v$  sebagai simpul dari  $G(A)$  dengan syarat  $0 \leq u, v \leq a_1 - 1$ . Graf  $G(A)$  mempunyai busur dari simpul  $u$  menunjuk ke simpul  $v$  jika dan hanya jika ada  $a_k$  yang merupakan elemen dari basis  $A$  dan memenuhi kekongruenan

$$u + a_k \equiv v \pmod{a_1};$$

(3)

sedemikian sehingga  $a_k$  merupakan bobot dari busur  $uv$ . Kemudian andaikan  $p$  merupakan lintasan pada graf  $G$  yang dimulai dari simpul 0 sampai simpul  $v$ ,  $e_i$  merupakan banyak busur yang memiliki bobot seharga  $a_i$  di dalam lintasan  $p$ , dan  $w$  merupakan total bobot dalam lintasan  $p$ . Untuk sembarang lintasan  $p$  yang dimulai dari simpul 0, maka dengan mengaplikasikan (3) didapatkan hubungan antara total bobot dengan simpul akhir lintasan.

$$v \equiv \sum_{i>1} e_i a_i = w \pmod{a_1}.$$

(4)

Ini berarti dengan asumsi  $\gcd(A) = 1$ ,  $G$  memiliki simpul-simpul yang terhubung kuat. Kita dapat buktikan dengan mencari lintasan dari  $u$  ke  $v$  dengan memilih nilai  $M$  yang besar sehingga  $v - u + M a_1 > f(A)$ . Kemudian kita dapat merepresentasikan bahwa  $G$  terhubung kuat dalam bentuk  $\sum_i e_i a_i = v - u + M a_1$  [Nijenhuis, 2005].

Terdapat korespondensi antara  $\{x_i\}$  dengan  $e_i$  apabila  $v \equiv$

$M \pmod{a_1}$  sehingga total bobot dari suatu lintasan :  $w = \sum_{i>1} e_i a_i \equiv v \pmod{a_1} \equiv M \pmod{a_1}$

Untuk  $w$  merupakan bobot dari lintasan dari simpul 0 ke  $v$ , dan dari (4)  $v$  kongruen dengan  $w \pmod{a_1}$ . Jika  $M \geq w$  maka  $M$  *representable* karena dapat dinyatakan dalam  $M = w \pmod{a_1}$  atau  $M = w + b \cdot a_1$  dengan  $b$  merupakan *integer* positif dan  $w = \sum_{i>1} e_i a_i$ ;  $e_i \geq 0$ . Sebaliknya, jika  $\sum_{i=1}^n a_i x_i = M$  maka  $\sum_{i>1} a_i x_i \equiv M \pmod{a_1} \equiv v \pmod{a_1}$ . Pada akhirnya hubungan antara  $w$ ,  $v$ , dan  $M$  dapat disimpulkan sebagai berikut.

$$w \leq \sum_{i>1} a_i x_i \leq M$$

(5)

Untuk mencari  $w$  terkecil yang mungkin kita harus memilih lintasan dengan bobot terkecil dalam graf  $G$ . Cara untuk menemukannya yaitu membangun sebuah pohon merentang minimum menggunakan Algoritma Prim atau Algoritma Kruskal. *Bilangan non-representable* terbesar yang kongruen dengan  $v \pmod{a_1}$  adalah  $w - a_1$  dengan  $w$  adalah bobot dari lintasan terpanjang di dalam pohon merentang minimum [Nijenhuis, 2005].

$$f(a_1, a_2, \dots, a_n) = D(G) - a_1.$$

(6)

Dengan  $D(G)$  merupakan bobot dari lintasan terpanjang di dalam pohon merentang minimum.

#### IV. PENERAPAN NIJENHUIS GRAPH UNTUK MEMECAHKAN MASALAH CHICKEN McNUGGET THEOREM

Dari konsep *Nijenhuis Graph* pada bab III akan dibuktikan apakah metode *Nijenhuis Graph* sesuai dengan teorema *Chicken McNugget*.

##### **Chicken McNugget Theorem:**

*All integers are McNugget numbers except 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 22, 23, 25, 28, 31, 34, 37, and 43. They are called 'non-McNugget numbers'. Thus the largest non-McNugget number is 43.* [Picciotto, 1994]

Pertama-tama masukkan ukuran kotak 6,9, dan 20 ke dalam basis  $A = \{6,9,20\}$ . Didapatkan  $a_1 = 6$ ,  $a_2 = 9$ , dan  $a_3 = 20$ . Maka terdapat 6 simpul yaitu  $V = \{0,1,2,3,4,5\}$ . Dari  $a_2$ ,  $9 \equiv 3 \pmod{6}$  dan dari  $a_3$ ,  $20 \equiv 2 \pmod{6}$ . Dengan menerapkan (3) pada bab III maka didapatkan 6 busur berbobot 9 yang menghubungkan simpul yang selisih nomornya 3 dan 6 busur berbobot 20 yang menghubungkan simpul yang selisih nomornya 2.

Setelah mendapatkan graf seperti pada gambar 4.1, selanjutnya kita harus mencari nilai  $w$  minimum. Caranya dengan membangun sebuah pohon merentang minimum. Baik Algoritma Prim maupun Algoritma Kruskal dapat digunakan. Keduanya mungkin akan menghasilkan pohon yang berbeda namun bobot yang sama seperti pada gambar 4.2.

## VI. UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan syukur *Alhamdulillah* kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan makalah ini dengan baik. Kemudian penulis juga mengucapkan terima kasih kepada orang tua penulis yang telah memberikan dukungan selama proses penulisan makalah ini. Penulis juga turut mengucapkan terima kasih kepada Dr. Judhi Santoso, M.Sc. selaku dosen mata kuliah IF 2120 Matematika Diskrit yang telah membimbing dan memberi materi kepada penulis selama proses pengajaran mata perkuliahan Matematika Diskrit.

## REFERENSI

- [1] Munir, Rinaldi. 2005. Matematika Diskrit Edisi ke-3. Bandung: Informatika.
- [2] Lehman, Eric and team. 2010. Mathematics For Computer Science. Massachusetts: MIT Lecturer Team.
- [3] Wah, Anita; Picciotto, Henri. 1994. "Lesson 5.8 Building-block Numbers". Algebra: Themes, Tools, Concepts p. 186.
- [4] D. Beihoffer; J. Hendry; A. Nijenhuis; S. Wagon. 2005. "Faster Algorithm for Frobenius Numbers," *Electronic Journal of Combinatorics*. submitted for publication, June 2005.
- [5] Math world Team. 2017. *Frobenius Number*. <http://mathworld.wolfram.com/FrobeniusNumber.html>. Diakses 1 Desember 2017.
- [6] Math world Team. 2017. *McNuggetNumber*. <http://mathworld.wolfram.com/McNuggetNumber.html>. Diakses 1 Desember 2017.
- [7] Western Association of Schools and Colleges. 2017. [http://artofproblemsolving.com/wiki/index.php/Chicken\\_McNugget\\_Theorem](http://artofproblemsolving.com/wiki/index.php/Chicken_McNugget_Theorem). Diakses 1 Desember 2017.

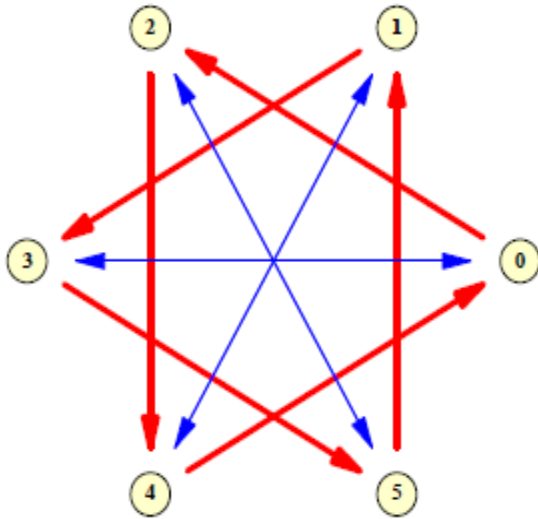
## PERNYATAAN

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.

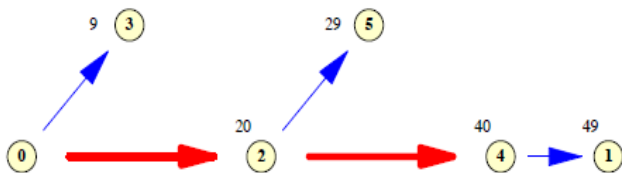
Bandung, 3 Desember 2017



Rabbi Fajar Mayoza-13516081



Gambar 4.1. Graf  $G(A)$  dengan  $A = \{6,9,20\}$ . Terdapat 6 simpul, 6 busur berwarna merah dengan bobot 20, dan 6 busur berwarna biru dengan bobot 9.



Gambar 4.2.. Pohon merentang minimum dari  $G(A)$  dengan  $A = \{6,9,20\}$ .

Dengan menjumlahkan harga-harga bobot lintasan terpanjang pada pohon merentang minimum kita mendapatkan nilai  $D(G) = 20 + 20 + 9 = 49$ . Jadi *Frobenius number*nya adalah  $D(G) - a_1 = 49 - 6 = 43$ . Nilai 43 selaras dengan *Chicken McNugget Theorem* yang menyatakan bahwa 43 merupakan bilangan *non-McNugget* terbesar.

## V. KESIMPULAN

Dapat disimpulkan bahwa banyak permasalahan yang dapat diselesaikan dan dioptimalkan solusinya menggunakan teori graf. Salah satu permasalahan yang dapat diselesaikan menggunakan teori graf adalah pencarian bilangan *non-McNugget* terbesar dalam *Chicken McNugget Theorem*. *Nijenhuis Graph* hadir sebagai solusi alternatif yang dinilai memiliki keunggulan yaitu lebih cepat daripada algoritma lain dalam menyelesaikan *Frobenius Equation* yang tidak dapat diselesaikan dengan rumus Sylvester. Seiring berjalannya waktu mungkin akan ditemukan solusi alternatif yang lebih optimal dan lebih mangkus mengingat masih banyak matematikawan yang meneliti serta mengembangkan metode dan algoritma untuk menentukan *Frobenius Number* hingga saat ini.