Lucas Theorem Untuk Mengatur Penyimpanan Memori yang Lebih Aman

Hendra Hadhil Choiri (135 08 041)

Program Studi Teknik Informatika ITB
Jalan Ganesha 10, Bandung
e-mail: hendra_h2c_mathematician@yahoo.com; if18041@students.if.itb.ac.id

ABSTRAK

Metode penyimpanan data dalam memori komputer sangat bervariasi. Salah satu alternatif adalah dengan menggunakan fungsi C(n,r) mod p. Untuk beberapa kasus memang nilainya dengan mudah dapat dihitung dengan cara yang umum. Namun untuk menangani kasus n dan r yang cukup besar, penghitungan C(n,r) tidak selalu mudah. Fungsi yang lumayan rumit ini dapat lebih membuat keamanan data terjaga. Dalam hal ini, Edouard Lucas berhasil menemukan metode berupa teorema yang dapat mempermudah penghitungan ini.

Kata kunci: Bilangan prima, Lucas theorem, kombinasi, penyimpanan data

1. PENDAHULUAN

Penyimpanan data adalah salah satu hal yang sangat erat hubungannya dengan matematika diskrit. Banyak metode yang dapat digunakan untuk menempatkan data-data yang disimpan di dalam memori komputer. Salah satu metode yang terkenal yaitu dengan menggunakan fungsi hash yang memiliki bentuk umum:

$$h(k) = k \bmod m \qquad (1)$$

di mana m adalah jumlah lokasi memori yang tersedia, dan k adalah kunci.

Memang sepertinya fungsi hash adalah fungsi yang cukup sederhana dan mudah dipahami. Namun, dalam beberapa kasus, sistem seperti ini keamanannya kurang terjaga dan datanya bisa dilacak oleh pihak-pihak yang tidak bertanggung jawab. Untuk itulah banyak usaha dilakukan dalam membuat metode untuk menempatkan data ke dalam memori komputer. Misalnya dengan fungsi:

$$f(a,b) = C(a,b) \bmod m \quad \text{,dengan } a \geq b \qquad (2)$$

mirip seperti fungsi hash, m menyatakan jumlah lokasi memori berbentuk sel-sel yang diberi indeks 0 sampai m-1). Hanya saja di sini, fungsinya membutuhkan dua kunci, yaitu a dan b, dengan a tidak lebih kecil dari b (agar dapat dihitung kombinasinya. Andaikan a≤b, tinggal dibalik nilainya). Dua buah kunci ini bisa berarti terdapat dua record yang dapat disimpan di memori, atau bisa juga kedua kunci disimpan di dua tempat berbeda demi keamanan, sehingga hanya orang-orang tertentu saja di mana saja record-record disimpan.

Jadi record dengan kunci a dan b akan disimpan di lokasi memori yang beralamat f(a,b). Fungsi yang digunakan memanfaatkan operasi kombinasi ($C(n,r)=\binom{n}{r}$) bertujuan agar penghitungan menjadi agak lebih rumit (dibanding dengan fungsi axb ataupun a^b). Dengan fungsi seperti itu penempatan memori menjadi sedikit lebih acak sehingga keamanannya lebih terjaga.

Permasalahan selanjutnya adalah bagaimana penghitungan fungsi tersebut. Memang sepertinya dapat diselesaikan dengan sederhana. Tinggal menghitung nilai C(n,r), lalu ubah menjadi bentuk km+q, dan kita dapatkan f(a,b)=q. Namun pada kenyataannya penghitungan C(n,r) tidak selalu mudah, misalnya dalam kasus n dan r yang cukup besar. Berbeda halnya dengan mencari nilai C(n,r) mod p. Dalam hal ini, seorang matematikawan bernama Edouard Lucas membuat teorema untuk menghitung nilai C(n,r) mod p, sebatas untuk p blangan prima. Teorema Lucas ini lah yang akan dibahas di makalah ini.

1.1 Bilangan Prima

Definisi: Bilangan bulat positif p yang lebih besar dari 1 disebut **bilangan prima** bila pembagi positif dari p (bilangan bulat) hanyalah 1 dan p. Bilangan bulat positif yang lebih besar dari 1 yang bukan prima disebut **bilangan komposit**.

Sesuai dengan namanya, bilangan-bilangan prima berperan sangat penting dan fundamental dalam Teori Bilangan. Dari definisi tersebut jelas bahwa 1 bukan bilangan prima meskipun pembagi positif dari 1 hanyalah dia sendiri. Juga jelas bahwa satu-satunya bilangan prima yang genap adalah bilangan 2.

Himpunan bilangan prima memang cukup unik karena tidak bisa dicari formula umumnya. Bahkan belum ada bukti kuat yang menyatakan batas dari bilangan prima, tapi dipercaya bahwa banyaknya bilangan prima adalah tak hingga. Namun, sudah banyak usaha diantaranya

tantangan untuk menemukan bilangan prima tertinggi yang berhasil ditemukan. Dalam dunia matematika banyak dibuat konjektur tentang bilangan prima.

Konjektur adalah suatu pernyataan atau dugaan yang secara matematis belum dapat dibuktikan kebenarannya maupun kesalahannya. Meskipun biasanya banyak sekali contoh kejadian yang membenarkan konjekur tersebut, tetapi bukti secara matematis dari konjektur tersebut belum diperoleh. Beberapa di antara konjektur bilangan prima antara lain:

- ✓ **Konjektur Goldbach**

Konjektur Goldbach yang diumumkan dalam tahun 1742 menyatakan bahwa **setiap bilangan bulat genap 2n yang lebih besar dari 4 merupakan jumlah dari dua bilangan prima yang ganjil.**

Dengan bantuan peralatan komputer, telah dapat diperlihatkan kebenaran konjektur Goldbach untuk semua bilangan bulat positif genap yang lebih kecil dari 4.1014. Bila bilangan genap 2n tersebut makin besar maka banyaknya cara penulisan untuk menyatakan 2n sebagai jumlah dari dua bilangan prima yang ganjil juga makin banyak. Sebagai contoh, terdapat 219.400 cara penulisan untuk bilangan genap 100.000.000. Meskipun konjektur Goldbach tampaknya benar, tetapi sampai sekarang belum ada bukti matematis dari konjektur tersebut.

Bila konjektur Golbach benar, maka **setiap bilangan bulat ganjil yang lebih besar dari 7 merupakan jumlah dari 3 bilangan prima yang ganjil**. Hal ini berdasarkan fakta bahwa bila n adalah bilangan bulat ganjil yang lebih besar dari 7 maka n = 3 + (n-3) dengan n-3 adalah bilangan bulat genap yang lebih besar dari 4. Selanjutnya dengan konjektur Goldbach n- 3 merupakan jumlah dari dua bilangan prima yang ganjil. Jadi n merupakanjumlah dari 3 bilangan prima yang ganjil.

- ✓ **Konjektur Bertrand**

Joseph Louis Francois Bertrand (1822-1900) ada1ah seorang matematikawan Perancis yang menda1ami Teori Bi1angan, Analisis, Geometri Diferensial clan Teori Probabilitas. Konjektur Bertrand yang diumumkan dalam tahun 1845, menyatakan bahwa antara bilangan bulat n ≥2 dan 2n terdapat sekurang-kurangnya satu bilangan prima. Meskipun dia tidak dapat membuktikan konjektur tersebut, tetapi dapat membuktikannya untuk semua bilangan-bilangan bulat n ≤3.000.000.

- ✓ **Konjektur Prima Kembar**

Dua bilangan prima disebut **Prima Kembar** bila selisih mereka adalah dua. Sebagai contoh adalah pasangan bilangan prima 3 clan 5, 5 clan 7, 11 dan 13, 101 dan 103, serta 4967 clan 4969. Konjektur Prima Kembar menyatakan bahwa terdapat tak berhingga banyak pasangan bilangan prima p clan p + 2. Dalam tahun 1966 matematikawan Cina, J. R. Chen, membuktikan bahwa terdapat tak berhingga banyak bilangan prima p sedemikian sehingga p + 2 punya paling banyak dua faktor prima.

- ✓ **Konjektur n^2+1**

Konjektur $n^2 + 1$ menyatakan bahwa terdapat tak berhingga banyak bilangan prima dengan bentuk $n^2 + 1$ di mana n adalah bilangan bulat positif. Bilangan prima terkecil dengan bentuk $n^2 + 1$ adalah $5 = 2^2 + 1$, kemudian $17 = 4^2 + 1$,$37 = 6^2 + 1$,$101 = 10^2 + 1$,$197 = 14^2 + 1$, dan seterusnya.

Dengan mengetahui konjektur-konjektur bilangan prima tersebut, diharapkan penggunaan teorema lucas dapat lebih optimal karena memang hanya berlaku untuk p bilangan prima.

1.2 Kombinasi

Kombinasi n dari r (n≥r) dilambangkan C(n,r) , ${}_nC_r$, atau $\binom{n}{r}$ didefinisikan sebagai:

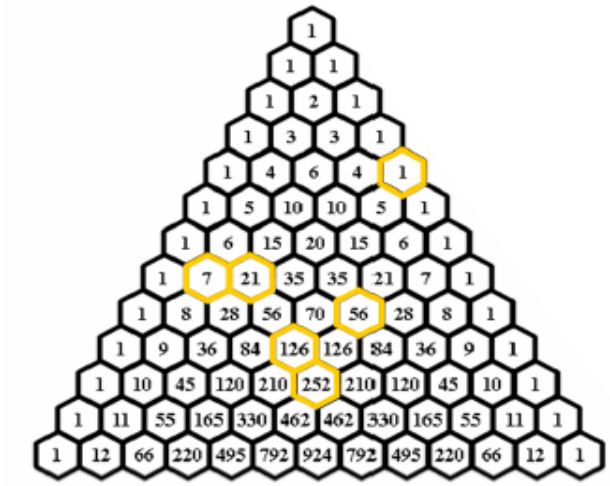
$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \qquad (3)$$

dengan n!=n.(n-1).(n-2).....2.1 dan 0!=1

Secara kombinatorik, operator ini berarti banyak cara memilih r benda dari n benda berbeda tanpa memperhatikan urutan.

Sedangkan dalam binomial, pada ekspansi $(x+y)^n$, $\binom{n}{r}$ menyatakan koefisien dari suku $x^r y^{n-r}$

Bilangan-bilangan hasil kombinasi ini tersusun rapi dalam segitiga pascal sebagai berikut:



Gambar 1. Segitiga Pascal

Pada segitiga Pascal, diawali dengan angka 1, lalu angka-angka di bawahnya merupakan penjumlahan dari 2 angka di atasnya. Pada segitiga Pascal, baris menyatakan n, dan dari kiri ke kanan di tiap baris merupakan r, sehingga tiap elemen segitiga Pascal adalah nilai dari $\binom{n}{r}$

1.3 Lucas Theorem

Setelah memahami prinsip dari operasi kombinasi, dan hal-hal yang berhubungan dengan bilangan prima, selanjutnya yaitu ini dari makalah ini, yaitu tentang teorema lukas / Lucas Theorem. Lucas theorem menyatakan bahwa:

Misal m dan n adalah bilangan bulat tak negatif, p bilangan prima, dan

$$m = m_k p^k + m_{k-1} p^{k-1} + \cdots + m_1 p + m_0,$$

serta

$$n = n_k p^k + n_{k-1} p^{k-1} + \cdots + n_1 p + n_0$$

(yang berarti $m_k m_{k-1} \ldots m_1 m_0$ dan $n_k n_{k-1} \ldots n_1 n_0$ menyatakan m dan n dalam basis p).

Maka berlaku hubungan:

$$\binom{m}{n} \equiv \prod_{i=0}^{k} \binom{m_i}{n_i} \pmod{p}, \tag{4}$$

Terdapat beberapa versi bukti dari lucas theorem. Salah satu bukti adalah memanfaatkan ekspansi binomial sebagai berikut:

Dasar utama dari pembuktian ini adalah fakta bahwa untuk suatu bilangan prima p dan suatu bilangan bulat r>0, $C(p^r, k)$ adalah kelipatan dari p untuk semua $0<k<p^r$.

Bukti dari pernyataan ini adalah bahwa untuk 0<k<N berlaku

$$\begin{aligned}\binom{N}{k} &= \frac{N!}{k!(N-k)!} \\ &= \frac{N(N-1)!}{k(k-1)!(N-k)!} \\ &= \frac{N}{k} \cdot \frac{(N-1)!}{(k-1)!((N-1)-(k-1))!} \\ &= \frac{N}{k} \cdot \binom{N-1}{k-1}\end{aligned}$$

Jadi, dengan mensubstitusikan N dengan p^r diperoleh:

$$k\binom{p^r}{k} = p^r \binom{p^r - 1}{k-1} \tag{5}$$

Yang berakibat p harus habis membagi $C(p^r,k)$.

Ketika k = 0 atau p^r, berakibat $C(p^r,k)=1$. Selain itu berlaku $C(p^r,k) \equiv 0 \pmod{p}$.

Kemudian, teorema binomial menunjukkan bahwa untuk p prima, berlaku:

$$(1+x)^{p\wedge r} \equiv 1 + x^{p\wedge r} \pmod{p} \tag{6}$$

(persamaan ini sering disebut "*Freshman Boinomial Theorem*").

Dengan persamaan tersebut, dapat dicari:

$$\begin{aligned}(1+x)^n = (1+x)^{\sum_{i=0}^{k} a_i p^i} &= \prod_{i=0}^{k} (1+x^{p^i})^{a_i} \\ &= \prod_{i=0}^{k} \sum_{b=0}^{a_i} \binom{a_i}{b} x^{bp^i}\end{aligned}$$

Jadi koefisien x^m di ruas kiri adalah C(n,m). Dan di ruas kanan nilainya sama dengan $\prod_{i=0}^{k} \binom{b_i}{a_i}$ di mana bi dan ai adalah digit ke-i dari n dan m di basis p. Hal ini sesuai dengan Lucas Theorem.

Sekarang mari kita lihat ilustrasinya.

Akan ditunjukkan bahwa $C(588,277) \equiv 54 \equiv 4 \pmod{5}$. Intinya akan dicari koefisien dari x^{277} pada ekspansi binomial $(1+x)^{588}$, akan kita cari dalam modulo 5. Padahal

$$(1+x)^{588} = (1+x)^{4(125)} (1+x)^{3(25)} (1+x)^{2(5)} (1+x)^3$$

Dan menurut persamaan (6), dalam modulo 5 nilai ini kongruen dengan

$$(1+x^{125})^4 (1+x^{25})^3 (1+x^5)^2 (1+x)^3$$

Dari bentuk ini, akan ditentukan banyak cara membuat suku dengan peubah x^{277}. Yaitu dengan memilih 2 suku x^{125}, sebuah x^{25}, 0 suku x^5 , and dua suku x^1. Dan banyak cara pemilihan adalah tepat $\binom{4}{2}.\binom{3}{1}.\binom{2}{0}.\binom{3}{2}$, yakni 6x3x1x3 = 54. Sehingga $C(588,277) \equiv 54 \equiv 4 \pmod{5}$.

(Q.E.D)

Dengan demikian, teorema ini valid dan bisa digunakan untuk menyelesaikan persoalan.

Dari teorema ini, ditemukan sebuah corollary yang cukup unik yaitu "Jika p prima dan N dinyatakan dalam basis p menjadi $(a_j,\ldots,a_1,a_0)$, maka ada tepat $(1+a_j)\ldots(1+a_1)(1+a_0)$ nilai k sehingga C(N,k) BUKAN merupakan kelipatan p.

1.4 Perbandingan

Sebuah teorema dibuat tentu saja untuk menyederhanakan masalah yang ada. Sebuah teorema tidak berguna jika permasalahan justru menjadi lebih rumit.

Permasalahan menghitung C(n,r) mod p bisa saja diselesaikan dengan cara konvensional, mencari nilai C(n,r) lalu dijadikan modulo p. Namun, seperti yang sudah diketahui, operasi kombinasi mengandung operasi faktorial (!). Dan untuk suatu n yang besar, menghitung n! Tidaklah mudah, sebagai bayangan, perhatikan tabel banyak digit n! Terhadap n berikut:

Tabel 1. Banyak digit n! Terhadap n

n	Banyak digit n!
1	1
2	1
3	1
4	2
5	3
6	3
7	4
8	5
9	6
10	7
20	19
30	33
40	48
50	65
60	82

Dari tabel di atas, perhatikan bahwa 60! saja menghasilkan bilangan dengan 82 digit. Bayangkan jika n mencapai ratusan. Mengalikan ratusan bilangan bulat bukan urusan yang mudah.

Memang penghitungan C(n,r) tidak mudah. Namun, beda hal jika yang dihitung adalah C(n,r) mod p. Cukup menyatakan n dan r dalam basis p, lau menerapkan teorema lukas. Penghitungan ini jauh lebih sederhana.

2. METODE

Setelah mempelajari bagaimana prinsip dari Lucas Theorem, dan melihat betapa bergunanya teorema ini untuk menyederhakan permasalahan, selanjutnya tinggal mengaplikasikan teorema ini dalam penyelesaian masalah. Misalnya untuk masalah penyimpanan data yang menjadi konsentrasi utama di makalah ini.

2.1 Analisis Kasus

Contoh kasus yang sederhana tentang penyimpanan data adalah sebagai berikut:

Demi keamanan, sebuah komputer menempatkan record-record ke memori dengan 2 kunci, a dan b. Komputer ini memiliki 13 buah lokasi memori (0 hingga 16) dengan aturan penempatan menggunakan fungsi $f(a,b) = C(a,b) \bmod 13$. Seorang pegawai akan menyimpan suatu record dengan kunci 208 dan 4796. Di lokasi memori manakah data tersebut akan disimpan? Diasumsikan semua lokasi memori masih kosong, sehingga pasti bisa ditempati.

Pada soal tersebut ada 2 kunci yaitu 208 dan 4796. Karena 4796>208, maka ditetapkan a=4796 dan b=208. Dan lokasi memori tempat data itu akan disimpan adalah $f(4796,208) = C(4796,208) \bmod 13$.

2.2 Penerapan Lucas Theorem

Dari permasalahan di 2.1, akan dicari nilai dari C(4796,208) mod 13. Akan diselesaikan dengan memanfaatkan lucas theorem. (Berhubung 13 adalah bilangan prima, Lucas Theorem berlaku).

Langkah pertama adalah dengan mengkonversikan kunci ke basis 13. Ada banyak metode untuk mengubah bilangan basis 10 ke basis-n. Untuk basis 13, digit-digit yang ada adalah 0..9, A=10, B=11, dan C=12. Perhatikan metode-metode berikut:

$$4796 : 13 = 368 \quad sisa \quad 12$$
$$368 : 13 = 28 \quad sisa \quad 4$$
$$28 : 13 = 2 \quad sisa \quad 2$$

Didapat bahwa $4796_{10} = 224C_{13}$

serta

$208 = 169 + 39 = 1x13^2 + 3x13^1 + 0x13^0$
sehingga $208_{10} = 130_{13}$

Selanjutnya, dengan menggunakan teorema lukas, didapat:

$$\binom{4796}{208} \equiv \binom{2}{0}.\binom{2}{1}.\binom{4}{3}.\binom{12}{0} \equiv 1.2.4.1$$
$$\equiv 8 \pmod{13}$$

Jadi, data akan disimpan di lokasi memori 8.

2.3 Kasus Khusus

Seperti yang sudah dijelaskan, **Lucas Theorem hanya menangani kasus modulo bilangan prima**. Padahal kenyatannya, lokasi penyimpanan **memori tidak selalu berupa bilangan prima**. Dan akan merepotkan jika harus mencari bilangan prima terbesar yang kurang dari itu. Sisa lokasi memori akan terbuang percuma.

Untuk menangani kasus modulo bilangan komposit, dapat dimanfaatkan **Chinese Remainder Problem**. Karena tiap bilangan komposit pasti merupakan hasil perkalian dari beberapa bilangan prima.

Misalnya pada contoh kasus 2.1. Sekarang komputernya sudah berkembang, memori lokasi memori yang tersedia ada 403, dan sistem penyimpanannya masih menggunakan fungsi yang sama. Berhubung 403=13x31, maka yang dapat dicari dengan Lucas Theorem adalah nilai dari C(4796,208) mod 13 dan C(4796,208) mod 31.

Nilai dari C(4796,208) mod 13 sudah ditemukan di bagian 2.2, yaitu 8. Selanjutnya akan dicari nilai dari C(4796,208) mod 31 dengan cara yang sama.

$4796 = 4x31^2 + 30x31 + 22$

$208 = 6x31 + 22$

Sehingga

$$\binom{4796}{208} \equiv \binom{4}{0}.\binom{30}{6}.\binom{22}{22} \equiv 1.\frac{30!}{6!24!}.1$$
$$\equiv \frac{24!.25.26.27.28.29.30}{2.3.4.5.6.24!} \pmod{31}$$
$$\equiv 25.\frac{26}{2}.\frac{27}{3}.\frac{28}{4}.29.\frac{30}{5.6} \pmod{31}$$
$$\equiv 25.13.9.7.29 \pmod{31}$$
$$\equiv 593775 \pmod{31}$$
$$\equiv 1 \pmod{31}$$

Jadi kita dapatkan 2 persamaan kongruensi:

C(4796,208) ≡ 8 (mod 13) → C(4796,208) = 13k+8..(7)
C(4796,208) ≡ 1 (mod 31) .....(8)

Dengan mensubstitusikan persamaan (7) ke (8) didapatkan:

$$13k + 8 \equiv 1 \pmod{31}$$
$$13k \equiv -7 \pmod{31}$$
$$13k.12 \equiv -7.12 \pmod{31} \quad [13^{-1} \bmod 31 = 12]$$
$$156k \equiv -84 \pmod{31}$$
$$k \equiv 9 \pmod{31}$$

Persamaan terakhir ekivalen dengan k=31h+9
Sehingga
C(4796,208) = 13k + 8
= 13(31h+9)
= 403h + 117
Yaitu C(4796,208)mod 403 = 117

Jadi record tersebut akan disimpan di lokasi **117**

Begitulah kira-kira proses penghitungan penentuan lokasi jika banyak memori yang ada berupa bilangan komposit. Sehingga Lucas Theorem dapat diperumum, mampu menangani kasus modulo bilangan komposit.

IV. KESIMPULAN

Matematika diskrit sangat besar kontribusinya dalam dunia sains dan teknologi. Misalnya dalam bidang teori bilangan. Banyak permasalahan dapat diselesaikan dengan teori bilangan, misal kriptogrfi, pengamanan data, kompresi, dan lain-lain.

Dalam hal penyimpanan data, banyak alternatif dan variasi yang bisa digunakan, misalnya dengan sistem fungsi hash, dan di makalah ini dijelaskan pendefinisian fungsi f(a,b) = *C*(a,b) *mod* m yang membutuhkan 2 buah kunci, a dan b. Fungsi yang lumayan rumit seperti ini dapat menambah tingkat keamanan data.

Untuk mempermudah penghitungan *C*(a,b) *mod* m, dapat digunakan Lucas Theorem yang menyatakan bahwa

Misal m dan n adalah bilangan bulat tak negatif, p bilangan prima, dan
$m = m_k p^k + m_{k-1} p^{k-1} + \cdots + m_1 p + m_0,$
serta
$n = n_k p^k + n_{k-1} p^{k-1} + \cdots + n_1 p + n_0$
(yang berarti $m_k m_{k-1} \ldots m_1 m_0$ dan $n_k n_{k-1} \ldots n_1 n_0$ menyatakan m dan n dalam basis p).
Maka berlaku hubungan:

$$\binom{m}{n} \equiv \prod_{i=0}^{k} \binom{m_i}{n_i} \pmod{p},$$

Berhubung Lucas theorem hanya berlaku untuk m bilangan prima, untuk menangani m bilangan komposit, digunakan Chinese Remainder Theorem dengan memecah m sebagai perkalian beberapa bilangan prima..

REFERENSI

[1] Munir, Rinaldi, "**Diktat Kuliah IF2091: Struktur Diskrit**", Penerbit ITB, 2008

[2] Smith, Douglas, "**A Great Theorem**", Miami University, 2007

[3] **"Mudd Math Fun Facts: Lucas Theorem",** URL: http://www.math.hmc.edu/funfacts/ffiles/30002.4-5.shtml diakses tanggal 18 Desember 2009 sore

[4] "**Wikipedia: Lucas' Theorem**", URL: http://en.wikipedia.org/wiki/Lucas%27_theorem diakses tanggal 18 Desember 2009 sekitar jam 15.00