PROSIDING

Sulistyo P

KONFERENSI NASIONAL MATEMATIKA XIV

Palembang, Kamis - Minggu, 24 s.d. 27 Juli 2008

Editor: Zulkardi

Matematika Adalah Jembatan Untuk Kehidupan Yang Lebih Baik

Diselenggarakan oleh:

limpunan Matematika Indonesia (IndoMS)

ekerjasama dengan

Jniversitas Sriwijaya

rogram Studi Magister Pendidikan Matematika,

rogram Pascasarjana

Kombinatorik

Statistika

PEMBANGKITAN PERMUTATION DENGAN SIKLUS

Sulistyo Puspitodjati [1] dan Djati Kerami[2]

[1]Jurusan Sistem Informasi FILKOM Universitas Gunadarma
E-mail: sulistyo@staff.gunadarma.ac.id
[2]Jurusan Matematika FMIPA Universitas Indonesia
E-mail: djatikr@ui.edu

Abstrak. Makalah ini membahas pembangkitan lengkap objek kombinatorial permutasi khususnya permutasi n dengan satu siklus dengan panjang n. Metode yang akan digunakan dalam pembangkitan permutasi dengan memperhatikan siklus tersebut menggunakan pendekatan pohon pembangkit atau metode ECO (*enumerating combinatorial objects*). Dalam metode ini setiap objek diperoleh dari objek yang lebih kecil dengan melakukan ekspansi lokal. Seringkali ekspansi lokal tersebut sangat teratur dan dapat dijelaskan dalam aturan suksesi. Metode ECO ini telah ditunjukkan efektif untuk beberapa struktur kombinatorik. Efektif dalam pembangkitan kombinatorik berarti: waktu untuk menghasilkan (running time) sebanding dengan banyaknya objek yang dihasilkan, yang merupakan syarat penting dalam merancang algoritma pembangkitan objek kombinatorial.

Kata kunci: *engbangkitan lengkap, permutasi dengan siklus, pohon pembangkit, metode ECO.*

1. Pendahuluan

Salah satu bidang utama dari kombinatorik adalah membangkitkan objek dari kelas tertentu untuk parameter tertentu, baik secara lengkap (*exhaustive generation*) atau secara acak (*random generation*). Maksud dari membangkitan secara lengkap tersebut adalah mencari cara atau metode atau algoritma untuk mencacah (*list, enumerate*) semua objek dalam urutan tertentu tanpa pengulangan dan tidak melewatkan satu objek pun. Algoritma-algoritma tersebut berguna pada banyak bidang seperti uji perangkat keras maupun perangkat lunak, biokimia, biologi dan termodinamika. ([Ber07], [Duc07]). Pembangkitan lengkap sering juga digunakan untuk memecahkan masalah-masalah NP-complete, dan menganalisa atau membuktikan suatu program. ([Vaj06]).

Salah satu pendekatan untuk membangkitkan objek kombinatorial adalah dengan yang disebut pohon pembangkit atau sering diidentikkan dengan nama metode ECO (*enumerating combinatorial objects*) ([Ban07]). Dalam metode ECO setiap objek diperoleh dari objek yang lebih kecil yang diekspansikan dengan rumusan yang disebut aturan suksesi. Aturan suksesi ini dapat direpresentasikan dalam suatu pohon dan disebut pohon pembangkit ([Fer05]). Pohon pembangkit ini telah ditunjukkan efisien dalam konteks pembangkitan kombinatorial, yaitu waktu untuk menghasilkan N objek berukuran n adalah O(N). Objek-objek yang telah ditunjukkan efisien dibangkitkan dengan pohon pembangkit tersebut adalah: objek Catalan dalam [Ber07] dan [Fer05], untuk permutasi penghindaran pola umum (*generelazid pattern avoidance*) dalam [Eli07], convex polyominoespan dalam [Lun03], dan untuk struktur Gray dalam [Ber207]. Pohon pembangkit juga secara detail dibahas untuk beberapa objek dalam [Ban07] dan [Wes96].

Selain itu, pohon pembangkit mempunyai pemanfaatan yang penting dalam kombinatorial, yaitu bijeksi dan pembangkitan acak ([Duc07]).

Karena itu makalah ini akan membahas pembangkitan permutasi siklus menggunakan pendekatan pohon pembangkit atau dikenal juga sebagai metode ECO

2. Permutasi

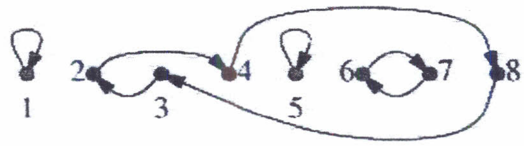
Permutasi adalah pemetaan dari suatu himpunan ke dirinya sendiri. Atau secara formal:

Definisi 1: Permutasi dari himpunan S = [n] = {1,2,..,n} adalah bijeksi $\pi: S \rightarrow S$.

Salah representasi dari permutasi adalah dengan perkalian siklusnya. Siklus dari permutasi adalah himpunan bagian dari suatu himpunan yang elemen-elemennya masuk dalam satu orbit. Atau siklus dengan panjang l dari suatu permutasi adalah urutan $a_1, a_2, \ldots, a_l$ sedemikian sehingga $a_i = \pi(a_{i-1})$ untuk i = 2, 3, …, l, dan $a_1 = \pi(a_l)$ atau $\pi^l(a_i) = a_i$. ([Rus03] dan [Bón02]). Contoh, permutasi π berikut:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 4 & 2 & 8 & 5 & 7 & 6 & 3 \end{pmatrix}$$

dapat diwakili oleh diagram berikut:



Permutasi tersebut mempuyai 4 siklus (1), (2 4 8 3), (5), and (6 7). (2 4 8 3) adalah siklus dengan panjang l = 4, karena $\pi^4(2) = 2$. Dengan perkalian siklus, π dapat dinyatakan sebagai $\pi = (1)(2483)(5)(67)$. Karena siklus (8324) menyatakan siklus yang sama dengan (2483), maka sering digunakan cara yang unik untuk menyatakan permutasi menggunakan notasi siklus, yang disebut sebagai **notasi siklus kanonikal**. Cara ini adalah menulis elemen terbesar pada setiap siklus terlebih dahulu, kemudian mengurutkan setiap siklus dari kecil ke besar berdasarkan elemen-elemen pertama pada siklus. Dengan demikian $\pi = (1)(2483)(5)(67)$ dalam notasi siklus kanonikal adalah $\pi = (1)(5)(76)(8324)$.

Banyaknya permutasi [n] dengan m siklus adalah bilangan Stirling tanpa tanda jenis pertama c(n,m) ([Rus03], [Bón02]). Dimana

$$\begin{aligned} &c(n,n) = 1, \\ &c(n,1) = (n-1)! \\ &c(n,m) = (n-1).c(n-1,m) + c(n-1,m-1), \quad 1 < m < n. \end{aligned} \tag{1}$$

Makalah ini akan membahas permutasi dengan m siklus, khusus m = 1, yang selanjutnya akan disebut sebagai permutasi n dengan siklus. Banyaknya permutasi n dengan siklus menurut formula (1) diatas adalah (n-1)!. Sehingga banyaknya permutasi 4 dengan siklus adalah 3! = 6, yaitu {(4123), (4312), (4132), (4213), (4231), (4321)}.

3. Pembangkitan Objek kombinatorial dan Pohon Pembangkit

Salah satu bidang dalam kombinatorik adalah pembangkitan objek secara lengkap. Pembangkitan ini berarti membangkitkan (menghadirkan) semua anggota dari kelas kombinatorial tertentu secara efisien yang sedemikian sehingga setiap anggota muncul tepat sekali. Salah satu pendekatan untuk pembangkitan lengkap adalah dengan yang disebut **pohon pembangkit**. Pohon pembangkit adalah pohon yang menggambarkan keluarga tertentu dari objek kombinatorial; tiap simpul berhubungan dengan satu objek, dan cabangnya menuju simpul yang mengkodekan alternatif yang dipilih dalam mengkonstruksikan objek. Pohon pembangkit menjanjikan komputasi yang cepat dalam mengenumerasi barisan objek. Metode pohon pembangkit ini disistematisasikan oleh Barcucci, Del lungo, Pergola, and Pinzani, dengan nama sistem ECO (*enumerating combinatorial objects*) ([Ban07]). Dalam metode ECO ini setiap objek diperoleh dari objek yang lebih kecil dengan melakukan ekspansi lokal. Seringkali ekspansi lokal tersebut sangat teratur dan dapat dijelaskan dalam aturan suksesi. Metode ECO ini telah ditunjukkan efektif untuk beberapa struktur kombinatorik, seperti: objek Catalan dalam [Ber07] dan [Fer05], untuk permutasi penghindaran pola umum (*generelazid pattern avoidance*) dalam [Eli07], convex polyominoespan dalam [Lun03], dan untuk struktur Gray dalam [Ber207]. Namun penelitian-penelitian tersebut belum membahas pembangkitan permutasi siklus dengan pohon pembangkit atau metode ECO tersebut. Subbab berikut akan menjelaskan metode ECO secara lebih rinci.

3.1 Metode ECO, aturan suksesi, dan pohon pembangkit

Bagaimana metode ECO bekerja dijelaskan dalam [Ber07], [Duc07], dan [Ban07], sebagaimana berikut.

Operator ECO

Misalkan $\mathcal{O}$ adalah kelas objek kombinatorial dan p: $\mathcal{O} \rightarrow$N adalah parameter yang hingga pada $\mathcal{O}$, yaitu parameter p sedemikian sehingga $\mathcal{O}_n = \{O \in \mathcal{O} : p(O) = n\}$ dari objek berukuran n adalah hingga. Misalkan $\vartheta : \mathcal{O} \rightarrow 2^{\mathcal{O}}$ adalah operator yang sedemikian sehingga $\vartheta(\mathcal{O}_n) \subseteq 2^{\mathcal{O}n+1}$. Operator ϑ menggambarkan bagaimana objek kecil menghasilkan objek yang lebih besar.

Proposisi 2-1: Jika ϑ memenuhi, untuk setiap $n \geq 0$,

1. untuk setiap $O' \in \mathcal{O}_{n+1}$, akan terdapat $O \in \mathcal{O}_n$ sedemikian sehingga $O' \in \vartheta$, dan
2. untuk setiap $O, O' \in \mathcal{O}_n$, akan menggambarkan $\vartheta(O) \cap \vartheta(O') = \emptyset$ kapanpun $O \neq O'$, maka famili himpunan $\mathcal{F}_{n+1} = \{ \vartheta(O): O \in \mathcal{O}_n \}$ adalah partisi dari $\mathcal{O}_{n+1}$.

Operator ϑ yang memenuhi kondisi 1 dan 2 tersebut di atas, dikatakan sebagai operator ECO. Jadi operator ECO membangkitkan semua objek $\mathcal{O}$ sedemikian sehingga setiap objek $O' \in \mathcal{O}_{n+1}$ diperoleh secara unik dari $O \in \mathcal{O}_n$. Operator ECO yang sedang melakukan ekspansi lokal pada objek yag disebut situs aktif dari objek. Operator ECO dapat digambarkan dengan pohon pembangkit, yaitu: pohon berakar yang simpu-simpulnya berhubungan dengan objek $\mathcal{O}$. Akar yang ditempatkan pada level 0 pada pohon, adalah objek dengan ukuran terkecil, m. Objek-objek dengan ukuran sama berada pada level yang sama dan anak dari objek O, adalah yang dihasilkan dari O melalui ϑ. Jika $\{|\mathcal{O}_n|\}_n$ adalah urutan yang ditentukan oleh banyaknya objek berukuran n, maka $f_{\mathcal{O}}(x) = \Sigma_{n\geq m} |\mathcal{O}_n| x^n$ adalah fungsi pembangkitnya.

Aturan suksesi

Aturan suksesi Ω adalah sistem $((a), \mathcal{P})$, mengandung aksioma (a) dan himpunan produksi atau aturan penulisan $\mathcal{P}$ didefinisikan pada himpunan label $M \subset N^+$:

$$\Omega = \begin{cases} (a) \\ (k) \quad \mapsto \quad (e_1(k))(e_2(k))\ldots(e_k(k)) \quad \text{untuk semua } k \in M \end{cases}$$

dimana $a \in M$ adalah nilai tertentu dan e_i adalah fungsi $M \rightarrow M$.

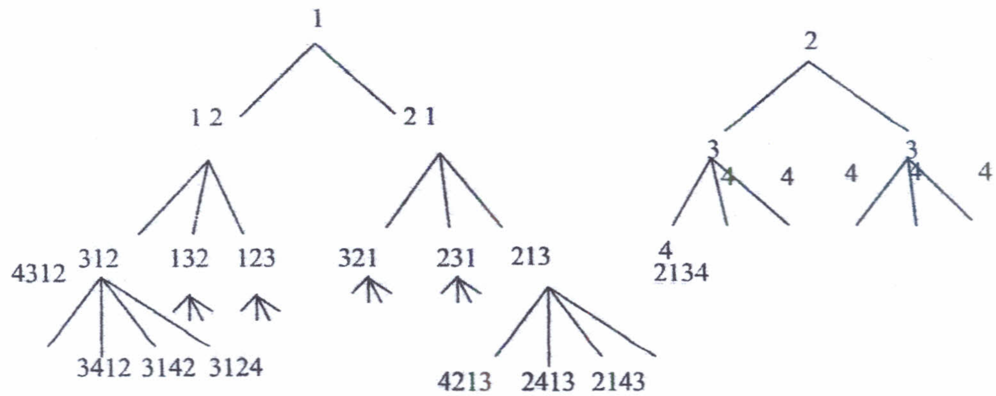
Salah satu sifat utama dari aturan suksesi adalah prinsip konsistensi, yaitu setiap label (k) harus memproduksi tepat k elemen. Aturan suksesi adalah sesuai dengan representasi pohon yang akarnya berlabel aksioma (a), dan simpul berlabel (k) menghasilkan level selanjutnya k anak-anak yang masing-masing berlabel $(e_1(k), \ldots, e_k(k))$ (yang nanti akan menghasilkan masing-masing anak berlabel $e_1(k), \ldots, e_k(k)$, dan seterusnya). Aturan suksesi menghasilkan urutan $\{f_n\}_n$ dari bilangan bulat positif, dimana f_n adalah banyaknya simpul pada level ke n dari pohon pembangkit dan dinotasikan sebagai $f_\Omega(x) = \Sigma_{n\geq m} f_n x^n$.

Seringkali operator ϑ dikodekan dengan aturan suksesi Ω, yang berarti, objek dengan ukuran minimum mempunyai a anak dan k objek $O'_1, \ldots, O'_k$, dihasilkan oleh objek O yang sedemikian sehingga O'_i akan menghasilkan anak $e_i(k)$ oleh ϑ, yaitu $|\vartheta(O'_i)| = e_i(k)$, $1 \leq i \leq k$. Berarti terdapat isomorfima antara pohon pembangkit dari operator ECO dan aturan suksesinya yang bersesuaian. Maka $f_{\mathcal{O}}(x) = x^m f_\Omega(x)$, atau $f_{\mathcal{O}}(x) = x^m f_\Omega(x)$ ketika $m = 0$.

3.2. Contoh : Permutasi

Pohon pembangkit untuk permutasi (secara umum) dapat dibangun berdasarkan algoritma pembangkitan permutasi Johnson –Trotter. Algoritma Johnson-Trotter memulai permutasi dari yang terpendek, yaitu [1], dan ini hanya mempunyai satu permutasi {1}. Kemudian untuk [2], elemen tambahan 2, ditambahkan ke permutasi 1 dengan cara meletakkan 2 pada sebelah kiri 1, atau ke sebelah kanan 1, sehingga diperoleh 12 dan 21. Dua elemen dari masing-masing permutasi ini mendefinisikan 3 posisi untuk elemen ketiga 3 dengan meletakkan 3 pada paling kiri, tengah, dan paling kanan: 312, 132, 123, dan 321, 231, 213. Secara umum, terdapat N cara untuk memperluas permutasi $a_1\ a_2\ \ldots\ a_{n-1}$ dengan panjang N-1 ke permutasi dengan panjang N:

$N\ a_1\ a_2 \ldots a_{n-1}$
$a_1\ N\ a_2 \ldots a_{n-1}$
$a_1\ a_2\ N \ldots a_{n-1}$
...
$a_1\ a_2 \ldots N\ a_{n-1}$
$a_1\ a_2 \ldots a_{n-1}\ N$



Gambar-1: Pohon pembangkit untuk permutasi

Melalui pohon pembangkit, pembangkitan permutasi dapat digambarkan sebagaimana terlihat pada gambar-1. Jika simpul dari pohon sebelah kiri pada gambar-1 diberi label sesuai dengan banyaknya anak cabang, maka diperoleh pohon sebelah kanan pada gambar-1. Dengan demikian pohon pembangkit untuk permutasi dapat ditulis dalam aturan suksesi berikut:

$$\Omega = \begin{cases} (1) \\ (k) \quad \mapsto \quad (k+1)^k \end{cases} \tag{3}$$

Berdasarkan aturan suksesi (3) maka fungsi pembangkit untuk aturan suksesi tersebut adalah

$$f_\Omega = \sum_{n \geq 0} f_n x_n = 1 + 2x + 2.3x^2 + 2.3.4x^3 + \ldots + (n+1)!x^n$$

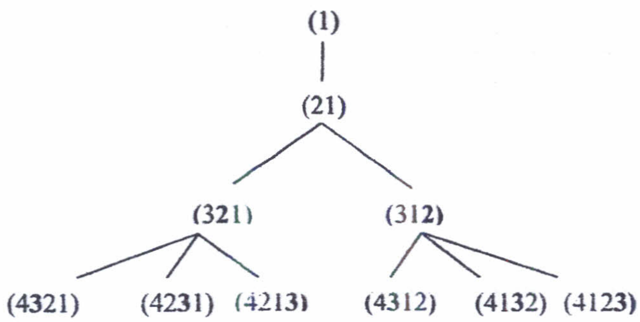
4. Pohon Pembangkit untuk Permutasi dengan siklus

Untuk pembangkitan permutasi π: [n]→ [n] dengan satu siklus panjang n, penulis mengikuti logika pembangkitan permutasi Johnson –Trotter. Misalkan S_{n_c} adalah himpunan semua anggota permutasi n dengan siklus yang ditulis secara kanonik, dan $|S_{n\_c}|$ adalah banyaknya anggota S_{n_c}. Maka

$$S_{n\_c} = \{(a_1a_2 \ldots a_n) |\ a_1 = n = \pi(a_n),\ a_i = \pi(a_{i-1}),\ i = 2, 3, \ldots, n\} \tag{4}$$

Dengan demikian, $\pi(n)$ adalah semua kemungkinan angota [n-1], kemudian $\pi(a_2)$ adalah semua anggota [n-1] tanpa a_2, dan seterusnya, $\pi(a_{n-1})$ adalah anggota [n-1] tanpa a_i, i = 2, …, n-2. Dengan kata lain siklus-siklus anggota S_{n_c} adalah $(na_2 \ldots a_n)$ dengan semua kemungkinan permutasi [n-1] untuk $a_2 \ldots a_n$. Berarti $|S_{n\_c}| = (n-1)!$

S_{n_c} dapat dibangun dari $S_{(n-1)\_c}$ dengan menambahkan n didepan semua anggota $S_{(n-1)\_c}$ sehingga diperoleh sebanyak $|S_{(n-1)\_c}|$ elemen berupa $(na_2 \ldots a_n)$ dengan $a_2 = n-1$. Anggota S_{n_c} yang lain dibentuk dari setiap $(na_2 \ldots a_n)$ yang ada dengan memindahkan posisi n-1 ke posisi a_k berturut-turut untuk k = 3, ..., n. Dengan demikian pohon pembangkit untuk permutasi n dengan siklus adalah sebagaimana pada gambar-2.



Gambar-2 Pohon Pembangkit untuk permutasi 4 dengan siklus

Jika simpul dari pohon pembangkit gambar-2 diberi label sesuai dengan banyaknya anak cabang, maka pohon pembangkit untuk permutasi n dengan siklus dapat ditulis dalam aturan suksesi berikut:

$$\Omega = \begin{cases} (1) & \\ (1) \mapsto & (2) \\ (k) \mapsto & (k+1)^k \end{cases} \tag{4}$$

Berdasarkan aturan suksesi (4) maka fungsi pembangkit untuk aturan suksesi tersebut adalah

$$f_\Omega = \sum_{n\geq 0} f_n x_n = 1 + x + 2.x^2 + 2.3x^3 + \ldots + n!x^n$$

5. Simpulan

Pohon pembangkit permutasi n dengan satu siklus panjang n pada makalah ini menunjukkan kekonsistensian dengan enumerasi tanpa pohon pembangkitan, yang menunjukkan bahwa pohon dapat digunakan untuk membangkitkan semua permutasi n dengan siklus.

Langkah selanjutnya yang perlu dikembangkan dari hasil pada makalah ini adalah analisa algoritma pembangkitan dengan pohon pembangkit ini untuk mengetahui apakah tahapan pembangkitan cukup efektif. Selain itu perlu diteliti lebih lanjut untuk melihat sifat-sifat yang dapat dihasilkan dari pohon ini, seperti kode gray misalnya.

Penelitian ini merupakan langkah sangat awal dalam penelitian untuk merumuskan pohon pembangkit permutasi untuk sembarang m siklus.

Acknowledgment

Terimakasih saya ucapkan pada Prof. Vajnovszki yang memberikan ide penelitian, dan tak lupa pada Pimpinan Universitas Gunadarma yang memungkinkan makalah ini dihadirkan pada Konferensi Matematika Nasional XIV 2008 Palembang.

Daftar Pustaka

[1] [Ban07] C. Banderier, dkk, *Generating Functions for Generating Trees*, arXiv:math.CO/ 0702753v1, 25 Feb 2007

[2] [Ber07] A. Bernini, I. Fanti, E Grazzini, *An exhaustive generation algorithm for Catalan objects and others*, arXiv:math.CO/0612127v2, 1 Feb 2007.

[3] [Ber207] A. Bernini, dkk, *A general exhaustive generation algorithm for Gray structures*, arXiv: math/0703262v1 [math.CO] , 9 March 2007

[4] [Bón02] M. Bóna, *A Walk through Combinatorics. An Introduction to Enumeration and Graph Theory*. New Jersey, War Scientific, 2002.

[5] [Duc07] E. Duchi, *ECO method and Object Grammars: two methods for the enumeration of combinatorial objects,* Dottorato di Ricerca in Ingegneria Informatica e dell'Automazione, XV Ciclo, Universit_a Degli Studi di Firenze, http://www.dsi.unifi.it/DRIIA/RaccoltaTesi/Duchi, diunduh: Mei, 2007

[6] [Eli07] S. Elizalde, *Generating Tree for Permutations Avoiding Generalized Patterns*, arXiv: 0707.4633v1 [math.CO] , 31 July 2007

[7] [Fer05] L. Ferrari and R. Pinzani, *Catalan like numbers and succession rules*, arXiv:math.CO/0507210v1, 11 July 2005.

[8] [Lun03] A. Del Lungo, dkk, "Enumeration of convex polyominoes using the ECO method", *Discrete Mathemathics and Theoretical Computer Science* AB(DMCS), 2003, 103-116.

[9] [Rus03] F. Ruskey, *Combinatorial Generation,* http://www.1stworks.com/ref/RuskeyCombGen.pdf, 2003.

[10] [Vaj06] V. Vajnovszki, *Generating Combinatorial Objects by ECO Method, the Lyndon Words Case,* Lecture Notes. Jakarta, 26 January 2006.

[11] [Wes96] J. West, *Generating Trees and Forbidden Subsequences* http://citeseerx.ist.psu.edu/showciting;jsessionid=1CEE9A0113F2F48EF4334245C6A52BED?cid=153114. 1996