

ALGORITMA PEMBANGKITAN MENGGUNAKAN POHON PEMBANGKIT

¹Sulistyo Puspitodjati

²Djati Kerami

¹UNIVERSITAS GUNADARMA (sulistyo@staff.gunadarma.ac.id)

²UNIVERSITAS GUNADARMA (djatiker@ui.edu)

ABSTRAK

Pembangkitan secara lengkap objek-objek dari kelas kombinatorial tertentu adalah mencari cara atau metode atau algoritma untuk mencacah (list, enumerate) semua objek dalam urutan tertentu tanpa pengulangan dan tidak melewatkan satu objek pun. Salah satu pendekatan dalam membangkitkan objek kombinatorial secara lengkap adalah dengan pohon pembangkit. Pohon pembangkit adalah suatu sistem yang mempunyai akar dan cabang-cabangnya yang dapat direpresentasikan dalam aturan yang dikenal dengan nama aturan suksesi. Pendekatan ini banyak digunakan karena dengan aturan suksesi dapat diterjemahkan kedalam bentuk-bentuk lain seperti operator linier pada polinomial dengan satu variabel, perkalian matriks, atau kode tertentu seperti kode Gray. Dari pohon pembangkit dapat pula dimungkinkan suatu algoritma pembangkitan acak. Makalah membahas pohon pembangkit dan aplikasinya pada objek kombinatorial untai Fibonacci, permutasi dan permutasi dengan siklus.

Kata Kunci: kombinatorik, pembangkitan lengkap, pohon pembangkit, aturan suksesi.

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 membangkitkan 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. TINJAUAN PUSTAKA

Enumerasi Objek Kombinatorik

Enumerasi (pembangkitan) kombinatorik adalah subjek utama dari kombinatorik dan menangani pencacahan (*counting*) sejumlah elemen dari kelas berhingga secara eksak maupun secara pendekatan. Berbagai masalah dari berbagai bidang dapat dipecahkan dari sudut pandang kombinatorial. Biasanya, masalah-masalah tersebut mempunyai ciri yang sama untuk direpresentasikan ke dalam objek yang sesuai untuk teknik-teknik pembangkitan kombinatorik. Duchi ([Duc07]), membagi pendekatan enumerasi dalam 4 kelompok, yakni: Fungsi rekurensi, fungsi pembangkit, DSV, dan metode ECO.

Untuk kelas objek O dan suatu parameter p dalam kelas tersebut, perhatikan himpunan O_n dengan ukuran n , dimana n adalah bilangan bulat tidak negatif. Yang dipermasalahkan dalam enumerasi kombinatorik adalah menentukan kardinalitas a_n dari himpunan O_n untuk setiap kemungkinan n . Hanya beberapa kasus yang mempunyai rumusan untuk a_n yang hanya melibatkan fungsi yang sudah diketahui dan bebas dari penjumlahan (*summation*). **Rekurensi** untuk a_n tersaji dalam hubungannya dengan nilai sebelumnya a_i yang telah dihitung. Dengan demikian memberikan prosedur sederhana untuk menghitung a_n untuk sembarang $n \in \mathbb{N}$.

Pendekatan lain adalah dengan **fungsi pembangkit**, perumusan a_n lebih umum, menggunakan deret Taylor, yaitu $f(x) = \sum_n a_n x^n$. Jika fungsi pembangkit sudah diketahui, maka koefisien a_n dapat diperoleh.

Metode lain untuk enumerasi adalah metode yang menggunakan **bahasa aljabar**, yang disebut metodologi Schützenberger, atau dikenal juga sebagai DSV. Metode ini menggunakan bijeksi antara objek-objek dan kata-kata dalam bahasa aljabar sedemikian rupa sehingga

nilai dari parameter objek berhubungan dengan panjang kata dari bahasa. Jika bahasa terbentuk dari tata bahasa bebas konteks yang ambigu, maka ada kemungkinan untuk menerjemahkan produksi dari tata bahasa ke sistem persamaan fungsional, yang solusinya unik dan *algebraic* dan merupakan fungsi pembangkit dari bahasa.

Pendekatan lain untuk enumerasi adalah dengan **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 Barucci, 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 sukseksi. 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 [Del03], dan untuk struktur Gray dalam [Ber207]. Berikut ini akan dijelaskan metode ECO secara lebih rinci.

Metode ECO dan aturan sukseksi

Metode ECO adalah metode untuk mengenumerasi kelas-kelas dari objek kombinatorial. Metode ini menawarkan konstruksi rekursif dari objek sesuai dengan ukurannya. Dalam metode ECO, setiap objek diperoleh dari objek yang lebih kecil dengan membuat beberapa ekspansi lokal. Jika konstruksi rekursif mempunyai peraturan tertentu, maka dapat

dikodekan dalam sistem formal yang disebut aturan sukseksi. Aturan sukseksi mewakili langkah selanjutnya. Bagaimana metode ECO bekerja dijelaskan dalam [Ber07], [Duc07], dan [Ban07], sebagaimana pada subbab-subbab berikut.

Operator ECO dan pohon pembangkit.

Misalkan \mathcal{O} adalah kelas objek kombinatorial dan $p: \mathcal{O} \rightarrow \mathbb{N}$ 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} .

$$\Omega = \left\{ \begin{array}{l} (a) \\ (k) \mapsto (e_1(k))(e_2(k)) \dots (e_k(k)) \end{array} \right. \text{ untuk semua } k \in M$$

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

Salah satu sifat utama dari aturan sukseksi adalah prinsip konsistensi, yaitu setiap label (k) harus memproduksi tepat k elemen. Aturan sukseksi adalah sesuai dengan representasi pohon yang akarnya berlabel aksioma (a) , dan simpul berlabel (k) menghasilkan level selanjutnya k anak-anak yang masing-masing berlabel $(e_1(k), \dots, e_k(k))$ (yang nanti akan menghasilkan masing-masing anak berlabel $e_1(k), \dots, e_k(k)$, dan seterusnya). Aturan sukseksi

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 yang disebut situs aktif dari objek. Operator ECO dapat digambarkan dengan pohon pembangkit, yaitu: pohon berakar yang simpul-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 $f_{\mathcal{O}}(x) = \sum_{n \geq m} |\mathcal{O}_n| x^n$ adalah fungsi pembangkitnya.

Aturan sukseksi

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

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 $f_{\Omega}(x) = \sum_{n \geq m} f_n x^n$.

Seringkali operator ϑ dikodekan dengan aturan sukseksi Ω , yang berarti, objek dengan ukuran minimum mempunyai a anak dan k objek O'_1, \dots, 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 isomorfisma antara pohon pembangkit dari operator ECO dan aturan suksesinya yang bersesuaian. Maka $f_{\vartheta}(x) = x^m f_{\Omega}(x)$, atau $f_{\vartheta}(x) = x^m f_{\Omega}(x)$ ketika $m = 0$.

Jika ϑ adalah operator ECO untuk \mathcal{O} sesuai p , dan asumsikan terdapat aturan sukseksi yang berhubungan dengan ϑ , maka kuadrupel $\Sigma = (\mathcal{O}, p, \vartheta, \Omega_{\vartheta})$ disebut sistem ECO.

3. APLIKASI

Berikut adalah beberapa penerapan pembangkitan objek dari beberapa kelas kombinatorial, yaitu: Untai

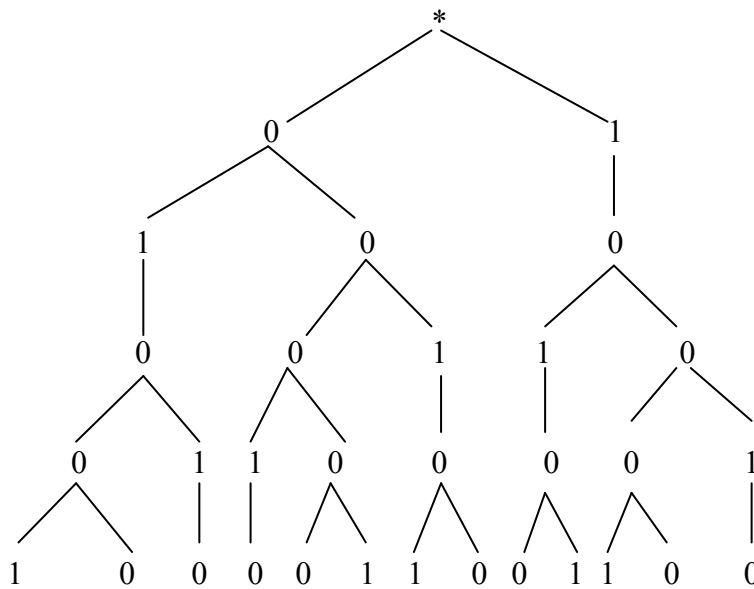
Fibonacci, permutasi, permutasi dengan siklus, dan jalur Dyck .

Pohon Pembangkit Untai Fibonacci

Untai Fibonacci adalah unta biner yang tidak mengandung 1 berurutan. Pohon pembangkit dari unta Fibonacci dapat dilihat pada gambar-1. Untai bit yang diperoleh dari pohon pembangkit tersebut adalah dengan membaca dari akar sampai ke simpul terbawah dimulai dari cabang kiri kemudian kanan.

Aturan sukseksi dari pohon pembangkit tersebut adalah:

$$\Omega = \begin{cases} (1) \\ (1) \mapsto (2) \\ (2) \mapsto (1)(2) \end{cases}$$



Gambar 1: Pohon pembangkit unta Fibonacci

Hal ini menunjukkan bahwa ukuran objek terkecil adalah $a = 1$. Dan selanjutnya terdapat simpul berlabel 1 yang mempunyai satu cabang anak berlabel 2, dan simpul berlabel 2 mempunyai 2 anak yang berlabel 1 dan 2.

Berdasarkan aturan sukseksi tersebut, maka fungsi pembangkit dari unta bit Fibonacci adalah $x/(1-x-x^2)$. ([Ban07]).

Pohon Pembangkit Permutasi

Permutasi $[n] = \{1,2,\dots,N\}$ dapat didefinisikan sebagai bijeksi $f: [n] \rightarrow [n]$. Pohon pembangkit permutasi 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

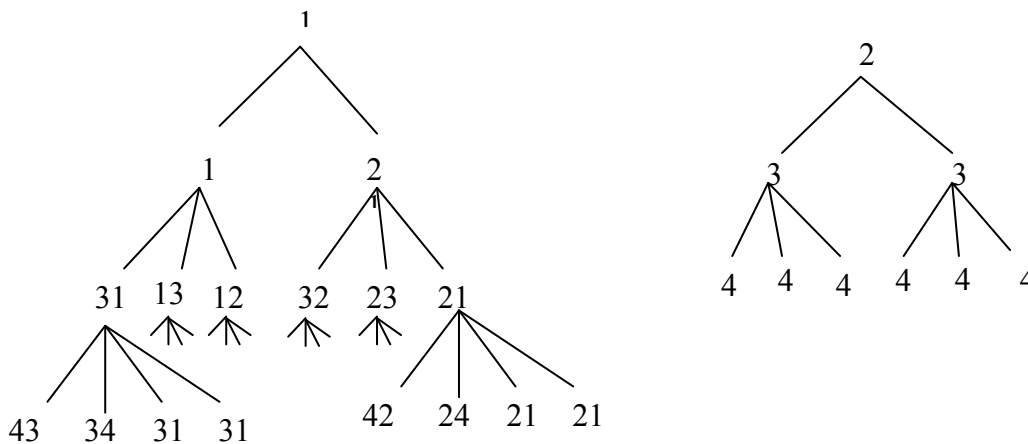
kiri 1, kemudian ke sebelah kanan 1, sehingga diperoleh 12 dan 21. Dua elemen dari masing-masing permutasi ini mendefinisikan 3 posisi untuk elemen ketiga 3 dengan meletakkan 3 pada paling kiri, tengah, dan paling kanan: 312, 132, 123, dan 321, 231, 213. Secara umum, terdapat N cara untuk memperluas permutasi $a_1 a_2 \dots a_{n-1}$ dengan panjang $N-1$ ke permutasi dengan panjang N :

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

$$a_1 a_2 \dots a_{n-1} N$$

Melalui pohon pembangkit, pembangkitan permutasi dapat digambarkan sebagaimana gambar-2 berikut. Jika simpul dari pohon sebelah kiri pada gambar-2 diberi label sesuai dengan banyaknya anak cabang, maka diperoleh pohon sebelah kanan pada gambar-2. Dengan demikian pohon pembangkit untuk permutasi dapat ditulis dalam aturan sukseksi berikut:

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



Gambar-2: Pohon pembangkit untuk permutasi

Berdasarkan aturan sukseksi untuk pohon pembangkit diatas, maka

fungsi pembangkit untuk aturan sukseksi tersebut adalah

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

Pohon Pembangkit untuk Permutasi dengan siklus

Permutasi dengan siklus adalah permutasi $(a_1 a_2 \dots a_n)$ dengan $a_1 = n = \pi(a_n)$, $a_i = \pi(a_{i-1})$, $i = 2, 3, \dots, n$. Untuk pembangkitan permutasi $\pi: [n] \rightarrow [n]$ 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

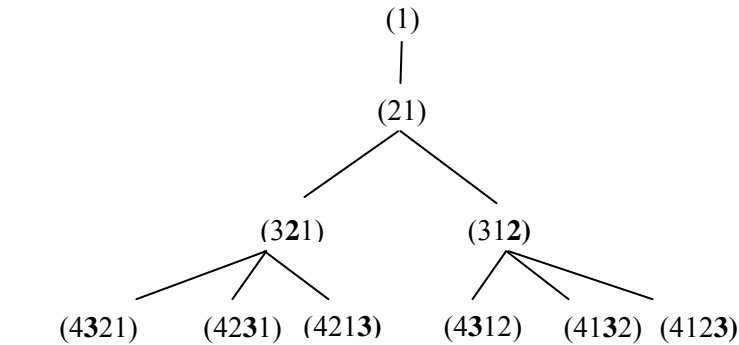
$$S_{n_c} = \{(a_1 a_2 \dots a_n) \mid a_1 = n = \pi(a_n), a_i = \pi(a_{i-1}), i = 2, 3, \dots, n\}$$

Dengan demikian, $\pi(n)$ adalah semua kemungkinan anggota $[n-1]$, 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 , $i = 2, \dots, n-2$. Dengan kata lain siklus-siklus anggota S_{n_c} adalah $(na_2 \dots a_n)$ dengan semua kemungkinan permutasi $[n-1]$ untuk $a_2 \dots a_n$. 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 $(na_2...a_n)$ dengan $a_2 = n-1$. Anggota S_{n_c} yang lain dibentuk dari setiap $(na_2...a_n)$ yang ada dengan memindahkan posisi $n-1$ ke posisi a_k berturut-turut untuk $k = 3, \dots, n$. Dengan demikian pohon pembangkit untuk

permutasi n dengan siklus adalah sebagaimana pada gambar-3.

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 Ω berikut:



Gambar-3 Pohon Pembangkit untuk permutasi 4 dengan siklus

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

Berdasarkan aturan suksesi diatas maka fungsi pembangkit untuk aturan suksesi tersebut adalah

$$f_{\Omega} = \sum_{n \geq 0} f_n x^n = 1 + x + 2 \cdot x^2 + 2 \cdot 3x^3 + \dots + n!x^n$$

4. KESIMPULAN DAN SARAN

Metode pembangkit menggunakan pohon pembangkit baik untuk untai Fibonacci maupun untuk permutasi, dilihat dari rumusan fungsi pembangkit untuk aturan suksesi sama dengan fungsi-fungsi pembangkit dari enumerasi yang sudah ada. Hal ini menunjukkan pembangkitan objek kombinatorial konsisten dengan yang sudah ada. Dan karena kelebihan yang disajikan dengan struktur pohon, maka pohon-pohon pembangkit diatas adalah pembangkitan yang baik untuk digunakan.

DAFTAR PUSTAKA

- [1] C. Banderier, dkk. 2007. *Generating Functions for Generating Trees*. arXiv:math.CO/0702753v1, 25 Feb 2007.
- [2] A. Bernini, I. Fanti, E. Grazzini. *An exhaustive generation algorithm for Catalan objects and others*. 2007. arXiv:math.CO/0612127v2, 1 Feb 2007.
- [3] A. Bernini, dkk. *A general exhaustive generation algorithm for Gray structures*. 2007. arXiv:math/0703262v1 [math.CO], 9 March 2007

- [4] M. Bóna, *A Walk through Combinatorics. An Introduction to Enumeration and Graph Theory*. 2002. New Jersey, War Scientific.
- [5] E. Duchi. *ECO method and Object Grammars: two methods for the enumeration of combinatorial objects*. 2007. 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>. diakses: Mei, 2007.
- [6] S. Elizalde. *Generating Tree for Permutations Avoiding Generalized Patterns*. 2007. arXiv: 0707.4633v1 [math.CO] , 31 July 2007.
- [7] L. Ferrari and R. Pinzani. *Catalan like numbers and succession rules*. 2005. arXiv:math.CO/0507210v1, 11 July 2005.
- [8] A. Del Lungo, dkk. "Enumeration of convex polyominoes using the ECO method". 2003. *Discrete Mathematics and Theoretical Computer Science* AB(DMCS), 2003, 103-116.
- [9] F. Ruskey. *Combinatorial Generation*. 2003. <http://www.1stworks.com/ref/RuskeyCombGen.pdf>.
- [10] V. Vajnovszki, *Generating Combinatorial Objects by ECO Method, the Lyndon Words Case*, Lecture Notes. Jakarta, 26 January 2006.
- [11] J. West. *Generating Trees and Forbidden Subsequences*. 1996. <http://citeseerx.ist.psu.edu/showciting;jsessionid=1CEE9A0113F2F48EF4334245C6A52BED?cid=153114>.