

## PEMBANGKITAN LENGKAP OBJEK CATALAN

<sup>1</sup>Sulistyo Puspitodjati

<sup>2</sup>Asep Juarna

<sup>1</sup>Universitas Gunadarma ([sulistyo@staff.gunadarma.ac.id](mailto:sulistyo@staff.gunadarma.ac.id))

<sup>2</sup>Universitas Gunadarma ([ajuarna@staff.gunadarma.ac.id](mailto:ajuarna@staff.gunadarma.ac.id))

### ABSTRAK

*Pembangkitan lengkap (exhaustive generation) obyek-obyek kombinatorial dalam sebuah kelas kombinatorial adalah salah satu pokok bahasan Kombinatorika. Berbagai metoda pembangkitan lengkap banyak dipakai dalam berbagai bidang seperti pengujian perangkat keras dan lunak, biologi, pembuktian algoritma, pembuktian teorema, dan lain-lain. Di dalam makalah ini akan dikemukakan salah satu metoda pembangkitan salah satu kelas kombinatorial yang dikenal sebagai Dyck Path. Dengan menggunakan metoda pohon pembangkit (generating tree) akan ditunjukkan bahwa untuk obyek-obyek berukuran  $n$ , kelas kombinatorial ini dicacah oleh bilangan Catalan ke- $n$ .*

*Selanjutnya di dalam makalah ini juga dikembangkan sebuah algoritma yang mendaftar (list) obyek-obyek yang sama sedemikian rupa sehingga setiap dua obyek yang terdaftar berurutan mempunyai perbedaan struktur yang kecil.*

*Kata Kunci: Pembangkitan lengkap, Dyck Path, bilangan Catalan ke- $n$ , pohon pembangkit*

### 1. PENDAHULUAN

Pembangkitan lengkap (*exhaustive generation*) suatu objek kombinatorial sering digunakan pada bidang-bidang yang memerlukan untuk menghadirkan objek-objek dengan urutan tertentu, seperti pada bidang: uji perangkat keras maupun lunak, biologi dan biokimia, atau termodinamika.

Salah satu pendekatan untuk membangkitkan objek kombinatorial secara lengkap adalah dengan yang disebut pohon pembangkit atau sering diidentikkan dengan nama metode ECO (*enumerating combinatorial objects*) ([Ban07]). Pohon pembangkit ini telah ditunjukkan efisien dalam konteks pembangkitan kombinatorial, yaitu waktu untuk menghasilkan  $N$  objek berukuran  $n$  adalah  $O(N)$ . ([Ber07], [Ber207], [Fer05], [Eli07], [Ban07] dan [Wes96]). Selain itu, pohon pembangkit mempunyai

pemanfaatan yang penting dalam kombinatorial, yaitu bijeksi dan pembangkitan acak ([Duc07]).

*Dyck Path* merupakan salah satu objek Catalan yang bijeksi terhadap banyak objek kombinatorial. ([Jua06]). Karena itu makalah ini menyajikan pohon pembangkitan dari objek kombinatorial *Dyck Path* yang mendaftar objek-objeknya sedemikian sehingga setiap dua obyek yang terdaftar berurutan mempunyai perbedaan struktur yang kecil.

### 2. TINJAUAN PUSTAKA

#### Definisi

Berikut adalah definisi-definisi atau istilah yang akan digunakan dalam makalah ini.

*Jalur (path)* adalah barisan titik anggota  $N \times N$  ( $N = \{0, 1, 2, \dots\}$ ).

*Langkah (step)* adalah pasangan dua titik berurutan dalam jalur.

*Jalur Dyck* adalah jalur  $D = \{ s_0, s_1, \dots, s_{2n} \}$  sedemikian sehingga  $s_0 = (0,0)$ ,  $s_{2n} = (2n, 0)$  dan hanya mempunyai langkah timur laut ( $s_i = (x,y)$ ,  $s_{i+1} = (x+1, y+1)$ ) atau tenggara ( $s_i = (x,y)$ ,  $s_{i+1} = (x+1, y-1)$ ), dan banyaknya langkah timur laut sama dengan banyaknya langkah tenggara.

$D_n$  adalah himpunan *jalur Dyck berukuran n*, yaitu jalur Dyck-nya mempunyai panjang  $2n$ , atau  $n$  langkah timur laut.

*Puncak (resp. lembah)* adalah titik  $s_i$  dengan langkah ( $s_{i-1}, s_i$ ) adalah langkah timur laut (tenggara) dan langkah ( $s_i, s_{i+1}$ ) adalah langkah tenggara (timur laut).

*Piramid*  $p_h$ ,  $\forall h \in \mathbb{N}$ , adalah barisan  $h$  langkah timur laut diikuti  $h$  langkah tenggara sedemikian sehingga jika ( $s_i, s_{i+1}$ ) adalah langkah timur laut pertama dan ( $s_{i+2h-1}, s_{i+2h}$ ) adalah langkah tenggara terakhir dalam barisan, maka  $s_i = (x, 0)$  dan  $s_{i+2h} = (x + 2h, 0)$ .

*Turunan (resp. tanjakan)* adalah barisan terakhir dari langkah-langkah tenggara (timur laut) dari suatu jalur Dyck dan diberi nomor berdasarkan urutan dari kanan (kiri) ke kiri (kanan). Titik akhir dari turunan terakhir adalah titik terakhir dari tanjakan terakhir.

*Tinggi*  $h(s_i)$  adalah titik yang ordinatnya  $s_i$  dan titik selanjut  $s_{i+1}$  dari suatu langkah timur laut ( $s_i, s_{i+1}$ ).

*Luas jalur* adalah jumlah dari titik yang tidak turun dari suatu titik dengan tinggi titik tersebut.

*Titik maksimum luas jalur*  $P^n_{max}$  adalah jumlah jalur dalam piramid  $P_n$ .

$P$  disebut *jalur aktif* jika dapat diperoleh jalur Dyck yang lain dengan melepas langkah awal dan akhir.

## Pohon Pembangkit dan Metode ECO

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.

### Operator ECO.

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  $\vartheta$  menggambarkan bagaimana objek kecil menghasilkan objek yang lebih besar.

**Proposisi\_1:** Jika  $\vartheta$  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  $\vartheta$  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 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  $\vartheta$ . 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 suksesi

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

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

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

Salah satu sifat utama dari aturan 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  $(e_1(k), \dots, e_k(k))$  (yang nanti akan menghasilkan masing-masing anak berlabel  $e_1(k), \dots, e_k(k)$ , 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  $f_{\Omega}(x) = \sum_{n \geq m} f_n x^n$ .

Seringkali operator  $\vartheta$  dikodekan dengan aturan suksesi  $\Omega$ , 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  $\vartheta$ , 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  $f_{\mathcal{O}}(x) = x^m f_{\Omega}(x)$ , atau  $f_{\mathcal{O}}(x) = x^m f_{\Omega}(x)$  ketika  $m = 0$ .

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

### 3. METODE PENELITIAN

Pembangkitan jalur Dyck dengan pohon pembangkit dibangun dengan pertama-tama mendefinisikan operator  $\vartheta$  yang memenuhi proposisi\_1.

Operator  $\vartheta$ :

1. Jalur pertama adalah  $P^n_{max}$
2. Lepas langkah jalur awal dan akhir, kemudian sisipkan satu puncak pada setiap titik dari turunan terakhir jalur yang diperoleh kecuali titik terakhir. Setiap sisipan menghadirkan jalur Dyck yang baru.
3. Untuk setiap jalur yang terbentuk, ulangi perlakuan tersebut sampai jalur-jalur aktif terbentuk dengan
  - 1) Melepas langkah awal dan akhir
  - 2) Menyisipkan puncak pada setiap titik dari turunan yang diperoleh setelah melakukan 1).

Operator  $\vartheta$  dalam pohon pembangkit adalah pohon  $D_n$  sebagai berikut:

1. Akar adalah  $P_{max}^n$  dan merupakan level nol.
2. jika  $X \in$  pohon  $D_n$  berada pada level  $k \geq 0$  maka  $Y \in \vartheta(X)$  adalah anak dari  $X$  dan berada pada level  $k+1$ .

Gambar-1 adalah adalah pohon pembangkit jalur Dyck  $D_4$ . Terlihat pada pohon, bagaimana semua anggota  $D_4$  dibangkitkan dari setiap jalur Dyck yang aktif.

$$\Omega = \begin{cases} (n-1, n-1) \\ (k, i) \end{cases} \mapsto (1,0)(2,1)\dots(i, i-1)(i+1,1-1)\dots(k, i-1)$$

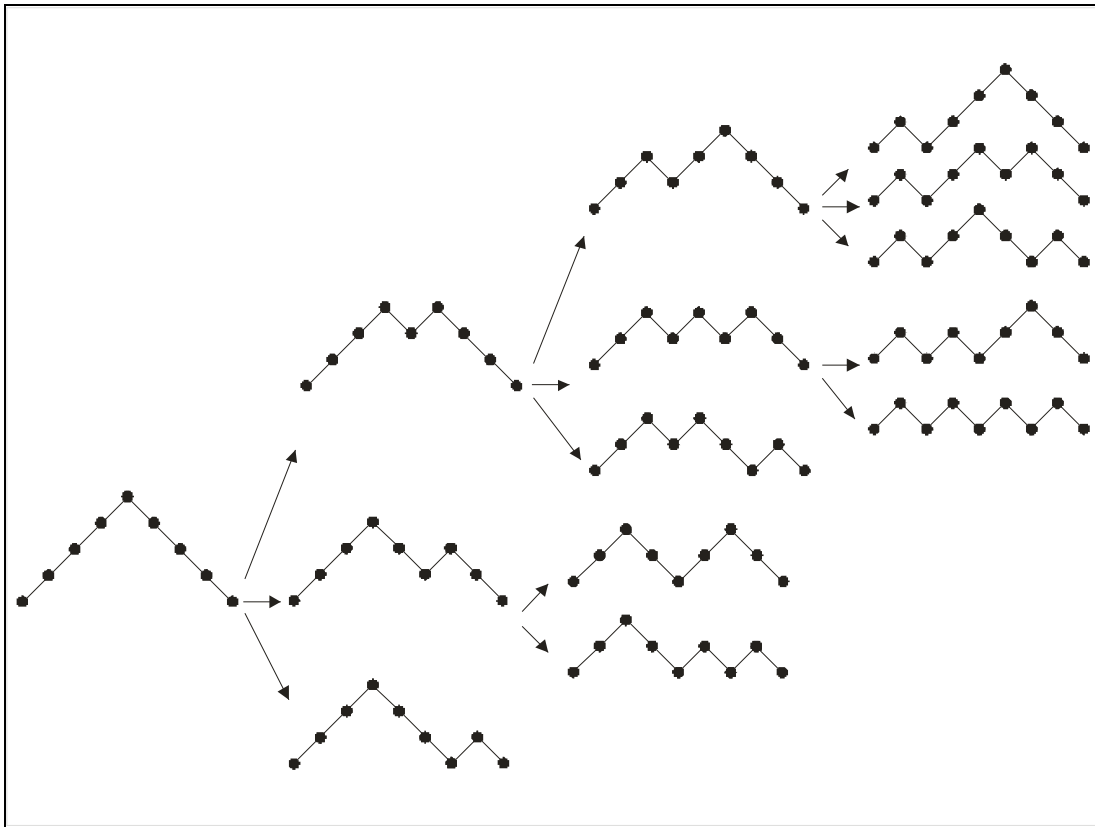
Jika diberi pohon pembangkit pada gambar\_1 diberi label dengan banyaknya anak (cabang) pada tiap level, maka aturan suksesi dapat dirumuskan menjadi

$$\Omega = \begin{cases} (n-1) \\ (k) \end{cases} \mapsto (2)(3)\dots(k) \text{ untuk semua } k \in M$$

#### 4. HASIL DAN PEMBAHASAN

##### Aturan Suksesi Pohon $D_n$

Hal terpenting dari membangun pohon pembangkit adalah rumusan aturan suksesi. Untuk pohon pembangkit  $D_n$ , aturan suksesi dapat dirumuskan dengan melabelkan pohon dengan  $(k,i)$  untuk menunjukkan banyaknya cabang  $k$  dengan tinggi dari puncak terendah adalah  $i$ . Dengan demikian, pohon mempunyai aturan suksesi berikut:



Gambar-1 Pohon Pembangkit jalur berukuran empat  $D_4$ .

### Algoritma Pembangkitan

Berdasarkan pengembangan pohon pembangkit atau berdasarkan operator yang sudah didefinisikan tersebut di atas, maka dibentuk algoritma pembangkitan yang dapat membangkitkan setiap objek sedemikian sehingga setiap objek dihasilkan dari objek terakhir yang dihasilkan. Pembangkitan dihubungkan dengan bagaimana simpul pohon dikunjungi, yaitu mengunjungi anak-anak dari simpul diurut berdasarkan jalur terpanjang dari turunan terakhir. Jadi anak  $P^n_{max}$  dibentuk dari piramid  $p_{n-1}$  diikuti oleh  $p_1$ .

### Algoritma 1

```

mulai dengan  $P^n_{max}$ 
bentuk anak pertama dari
membalik puncak  $P^n_{max}$ 
 $P := \text{anak pertama } P^n_{max}$  ;
while  $P \neq \text{anak terakhir dari } P^n_{max}$ 
do
  if memungkinkan then
     $P' := \text{op1}(P)$ 
  else if memungkinkan then
     $P' := \text{op2}(p)$ 
  else
     $P' := \text{op3}(P)$ 
  end if;
   $P := P'$ ;
end while
    
```

dimana:

- operasi op1 adalah membuang jalur pada awal dan akhir langkah, kemudian menyisipkan puncak pada titik terakhir dari turunan terakhir.

- operasi op2 adalah membalik puncak terkanan pada jalur.
- operasi op3 adalah buang  $p_1$  terkanan, kemudian sisipkan langkah timur laut pada awal dari jalur, dan satu langkah tenggara pada tanjakan terakhir kedua.

Dapat dilihat bahwa ketiga operasi membutuhkan sejumlah konstant aksi tidak tergantung dari panjang jalur, dengan demikian algoritma-1 adalah algoritma CAT (*constant amortized time*).([Rus03])

## 5. KESIMPULAN DAN SARAN

Makalah ini telah menyajikan pembangkitan semua dan hanya jalur-jalur Dyck  $D_n$  berukuran  $n$ . Algoritma yang disajikan dikembangkan adalah algoritma yang CAT karena hanya memerlukan sejumlah konstan komputasi per objek. Algoritma dapat dikembangkan untuk kelas kombinatorial yang bijeksi terhadap kelas jalur Dyck ini, seperti kelas tertentu dari permutasi.

## DAFTAR PUSTAKA

- [1] [Ban07] C. Banderier, dkk. 25 Feb 2007. *Generating Functions for Generating Trees*. arXiv:math.CO/0702753v1.
- [2] [Ber07] A. Bernini, I. Fanti, E. Grazzini. 1 Feb 2007. *An exhaustive generation algorithm for Catalan objects and others*. arXiv:math.CO/0612127v2.
- [3] [Ber207] A. Bernini, dkk. 9 March 2007. *A general exhaustive generation algorithm for Gray structures*. arXiv: math/0703262v1 [math.CO].
- [4] [Bón02] M. Bóna. 2002. *A Walk through Combinatorics. An Introduction to Enumeration and Graph Theory*. New Jersey, War Scientific.
- [5] [Duc07] E. Duchi, diunduh: Mei, 2007. *ECO method and Object Grammars: two methods for the enumeration of combinatorial object*. Dottorato di Ricerca in Ingegneria Informatica e dell'Automazione, XV Ciclo, Università Degli Studi di Firenze. <http://www.dsi.unifi.it/DRIIA/RaccoltaTesi/Duchi>.
- [6] [Eli07] S. Elizalde. 31 July 2007. *Generating Tree for Permutations Avoiding Generalized Patterns*. arXiv: 0707.4633v1 [math.CO].
- [7] [Fer05] L. Ferrari and R. Pinzani. 11 July 2005. *Catalan like numbers and succession rules*. arXiv:math.CO/0507210v1.
- [8] [Jua06] A. Juarna. July 2006. *Combinatorial Isomorphism Analysis On Some Extensions Of A Simion-Schmidt's Bijection*. Dissertation in Informatics, LE2I – U.F.R des Science des Technique, Université de Bourgogne,
- [9] [Rus03] F. Ruskey. 2003. *Combinatorial Generation*.

<http://www.1stworks.com/ref/RuskeyCombGen.pdf>.

- [10] [Wes96] J. West. 1996. *Generating Trees and Forbidden Subsequences*. <http://citeseerx.ist.psu.edu/showciting;jsessionid=1CE9A0113F2F48EF4334245C6A52BED?cid=153114>