
MENGKONSTRUKSI *TREE* DENGAN MENGGUNAKAN KODE DANDELION

Suci Nur Amalia dan Akmal Junaidi
Jurusan Matematika, FMIPA, Universitas Lampung

ABSTRAK

Tree merupakan salah satu bentuk graf yang banyak digunakan untuk merepresentasikan masalah dalam kehidupan sehari-hari, terutama dalam desain jaringan. Untuk merepresentasikan suatu *tree* terdapat beberapa teknik atau kode yang digunakan antara lain dengan kode Prufer, kode blob, kode grey, kode happy, dan kode dandelion. Dalam tulisan ini akan didiskusikan bagaimana cara mengkonstruksi suatu *tree*, khususnya *tree* berarah dari kode dandelion.

Kata-kata kunci: *tree*, *tree* berarah, kode dandelion.

PENDAHULUAN

Teori graf banyak memainkan peranan yang penting (Caccetta dan Vijayan (1987), Ahuja et al (1993). Suatu graf $G = (V, E)$ terdiri dari suatu himpunan V yang tidak kosong yang terdiri dari titik (*vertex*) dan suatu himpunan E yang mungkin kosong yang terdiri dari sisi (*edge*) yang menghubungkan titik-titik di V (Deo, 1989). Salah satu struktur graf adalah *tree* yang banyak dipakai dalam masalah desain jaringan, dan suatu *tree* dimana terdapat satu *vertex* dikhususkan dari semua *vertex* lainnya dinamakan *rooted tree*.

Untuk menyelesaikan masalah yang telah didesain dengan graf tadi diperlukan alat bantu komputer, terutama jika jaringan yang direpresentasikan tersebut merupakan jaringan yang mempunyai titik (*vertex*) yang banyak (Nandang, 2004). Untuk itu, diperlukan adanya teknik coding yang merubah bentuk *tree* ke suatu kode tertentu dan mengkonstruksi kembali suatu *tree* dari suatu kode tertentu yang didapat. Dalam tulisan ini akan didiskusikan bagaimana mengkodekan suatu *tree* berarah dengan kode dandelion, dan mengkonstruksikan kembali *tree* berarah dari suatu kode dandelion.

Tulisan ini dibagi atas tiga bagian : Bagian pertama berisi pendahuluan, pada bagian kedua akan dijelaskan teknik coding *tree* berarah dengan kode dandelion, pada bagian ketiga akan didiskusikan bagaimana cara mengkonstruksikan suatu *tree* berarah jika diberikan suatu kode dandelion, dan pada bagian keempat berisi kesimpulan.

Pengkodean Tree Bearah Dengan Kode Dandelion. Pembentukan kode dandelion dapat dilakukan dengan dua cara yaitu dengan pembentukan spanning tree dan dengan menggunakan permutasi siklik. Suatu *tree* yang memuat n *vertex* dengan label $(0, 1, 2, \dots, n)$ dan *edge* yang menghubungkan setiap dua *vertex* memiliki bobot $B_{succ(v_i)}$. Vertex 0 (nol) merupakan *root* dalam *tree* tersebut. $B_{succ(v_i)}$ menggambarkan *edge* dari *vertex* v_i sebagai *tail* ke *vertex* $succ(v_i)$ sebagai *head*. $B_{succ(v_i)}$ memudahkan dalam proses pengkodean dan rekonstruksi *tree* berarah. $B_{succ(v_i)}$ digunakan untuk mewakili setiap *edge* awal yang menghubungkan v_i ke $succ(v_i)$ yang kemudian digantikan dengan *edge* menghubungkan v_i ke 1. Selain itu, permutasi dan notasi siklik juga dapat mempermudah dalam proses tersebut. Notasi siklik mendefinisikan *path* dari *vertex* v_i ke v_n pada *tree*. Permutasi mendefinisikan setiap *edge* yang menghubungkan pasangan *vertex* v_i ke $succ(v_i)$. Pada kode dandelion, karakteristik *successor* setiap *vertex* (selain 0 dan 1) dengan bobot $B_{succ(v_i)}$ adalah 1 dan *successor* 1 adalah 0 (nol).

Langkah-langkah pembentukan *spanning tree* pada kode *Dandelion* adalah sebagai berikut:

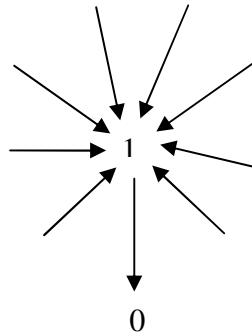
1. Pada langkah i , hapus $edge\ n-i+1 \rightarrow succ(n-i+1)$ dan tambahkan $edge\ n-i+1 \rightarrow 1$ dengan bobot $B_{succ(n-i+1)}$. (n didefinisikan banyaknya *vertex* selain *rooted 0*)
2. Apabila selama proses berlangsung terjadi *siklis*, maka:

Hapus $edge\ 1 \rightarrow succ(1)$ dan $edge\ n-i+1 \rightarrow 1$,

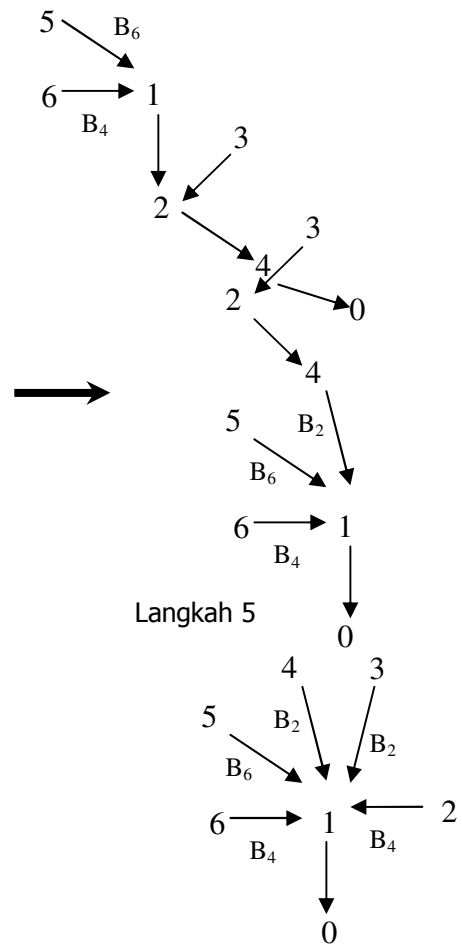
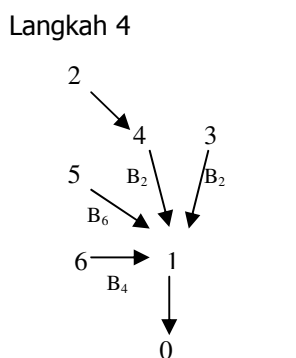
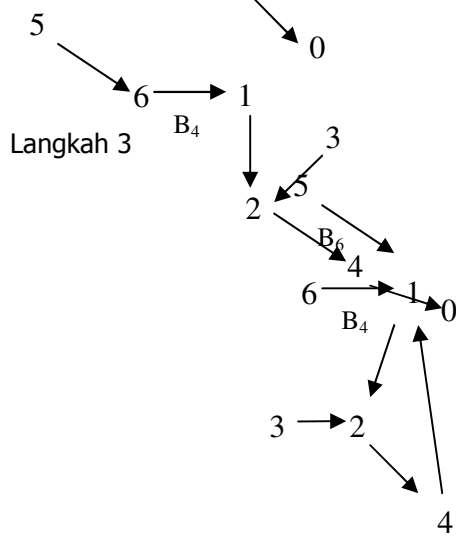
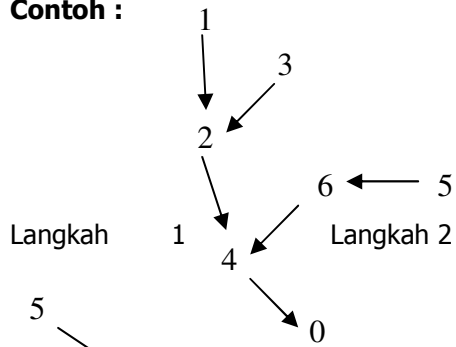
Gantikan $edge\ 1 \rightarrow succ(n-i+1)$ dan $edge\ n-i+1 \rightarrow 1$ (beri bobot $B_{succ(1)}$).

Jika tidak terjadi *siklis* selama proses berlangsung, maka kembali pada langkah 1.

3. Proses akan selesai pada langkah $i = n-1$, yang ditunjukkan pada gambar berikut:



Contoh :

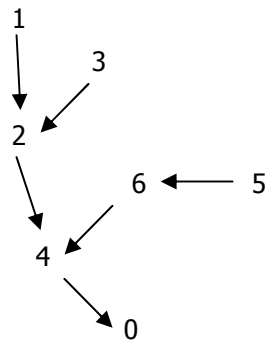


Didapat kode Dandelion: $(B_4, B_6, B_2, B_2, B_4)=(4,6,2,2,4)$.

Langkah-langkah permutasi dan notasi siklik untuk membentuk tree pada Kode Dandelion (Piccioto, 1999):

1. Menentukan path(1) dari *tree*.
2. Memeriksa setiap elemen *path*. Pada *step i* periksa apakah elemen (p_i) lebih besar dari semua elemen $path(p_i > p_{i+1} > \dots > p_n)$.
 - a. Jika iya, maka tulis dalam bentuk notasi siklik (p_1, p_2, \dots, p_i) dan kembali pada langkah (2). Proses berhenti ketika $i=n$, dimana n adalah banyaknya elemen pada *path*.
 - b. Jika tidak, maka kembali pada langkah (2).
3. Menuliskan semua notasi siklik yang diperoleh pada langkah (2) dalam bentuk permutasi.
4. Menuliskan *successors vertex* lainnya yang tidak dalam path(1) dalam notasi permutasi secara berurutan.
5. Kode diperoleh dari baru setiap vertex dalam notasi permutasi :
Code=(succ(2),succ(3),...succ(n))

Untuk memberikan gambaran tentang pembentukan kode Dandelion dengan permutasi siklik perhatikan *tree* berikut :



Langkah 1
Path(1) adalah (2,4).

Langkah 2.

4 merupakan *elemen* terbesar dalam path dan membentuk *siklis* permutasi : $\begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix}$

Langkah 3. Pada Tree tersebut hanya mengandung sebuah *siklis* permutasi pada path yang dibentuk.

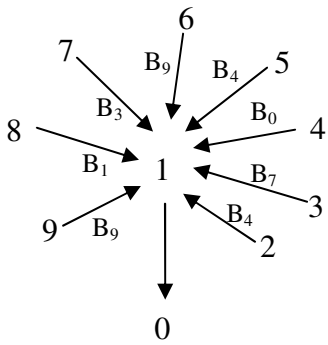
Langkah 4. Menuliskan semua verteks dalam notasi permutasi:
Dengan demikian kode Dandelion untuk *tree* tersebut adalah: (4,2,2,6,4).

$$(2,4) \rightarrow \begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 2 & 2 & 6 & 4 \end{pmatrix}$$

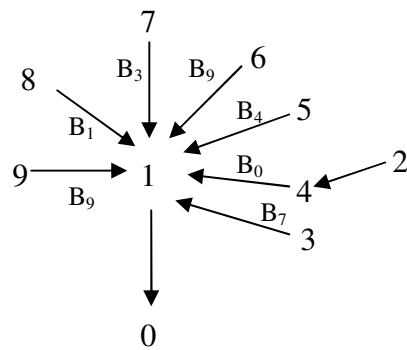
Mengkonstruksi Tree berarah dari Kode dandelion .Langkah-langkah membentuk *tree* berarah dari suatu kode *Dandelion* dari Piccioto (1999) sebagai berikut:

1. Konstruksi kode ke dalam bentuk graf dengan setiap *vertex* $(2, 3, \dots, n)$ menuju 1 dan bobot setiap edge $(2 \rightarrow 1, 3 \rightarrow 1, \dots, n \rightarrow 1)$ adalah $B_{succ(2)} B_{succ(3)}, \dots, B_{succ(n)}$.
2. Pada langkah i , hapus edge $i+1 \rightarrow 1$ dan bobot $B_{succ(n-i+1)}$, kemudian tambahkan edge $i+1 \rightarrow succ(i+1)$.
3. Apabila selama proses berlangsung menimbulkan *siklis* maka:
Hapus *edge* $1 \rightarrow succ(1)$ dan $i+1 \rightarrow succ(i+1)$, kemudian tambahkan *edge* $1 \rightarrow succ(i+1)$ dan *edge* $i+1 \rightarrow succ(1)$. Jika tidak terjadi siklis, maka lanjutkan langkah i .

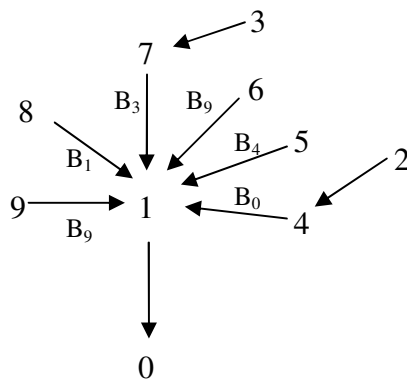
4. Proses akan selesai pada langkah $i=n-1$. (n adalah banyaknya vertex, selain 0(nol)). Untuk memberikan ilustrasi proses tersebut perhatikan contoh berikut :Misalkan diberikan kode Dandelion :(4,7,0,4,9,3,1,9)



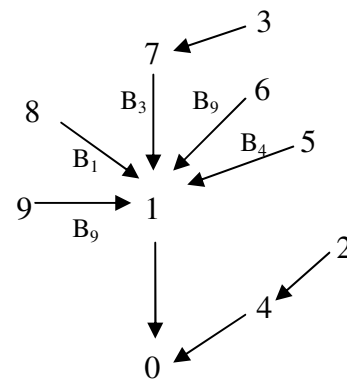
Langkah 1



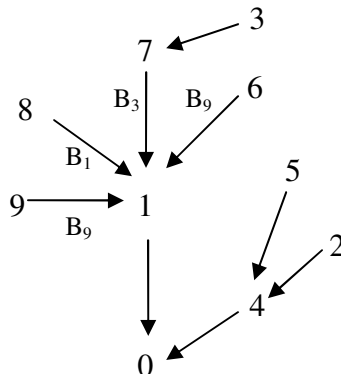
Langkah 2

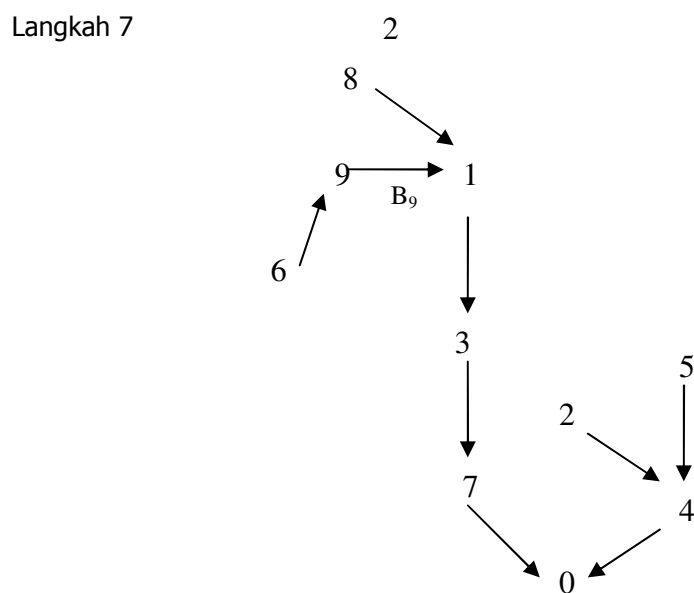
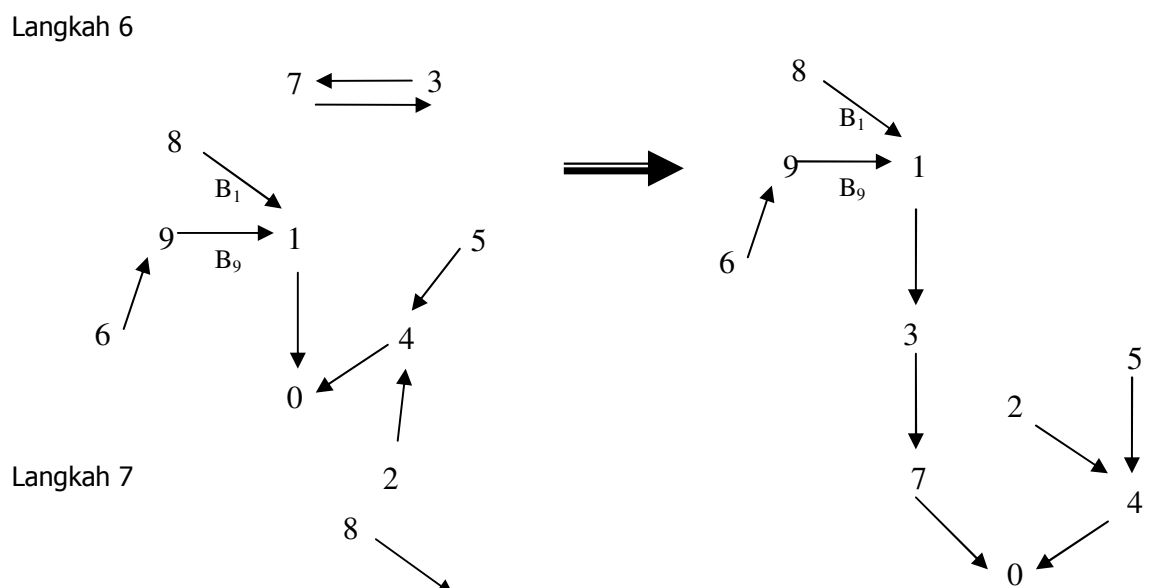
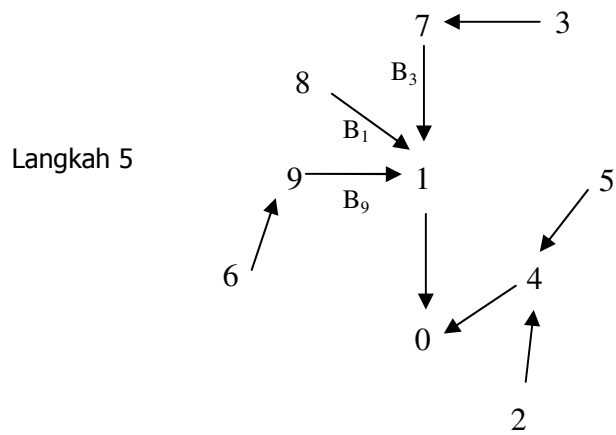


Langkah 3

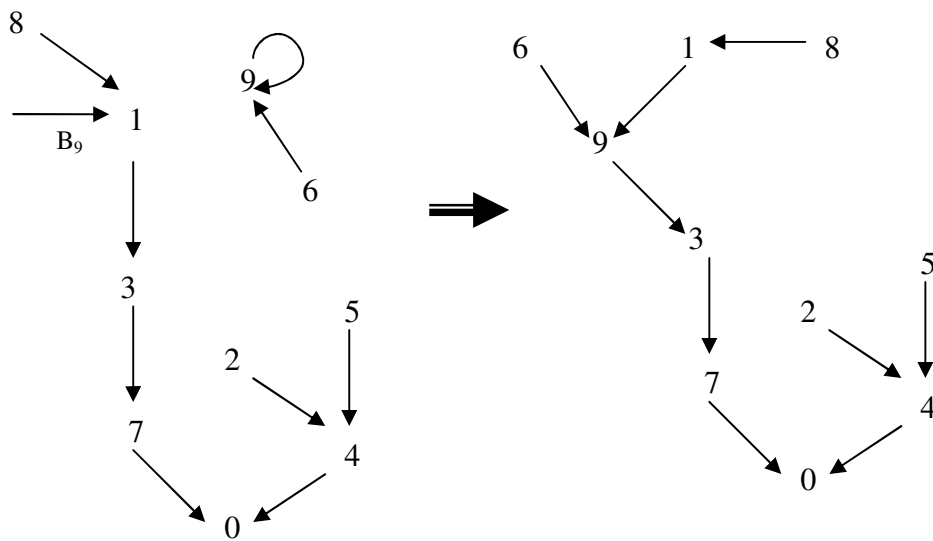


Langkah 4





Langkah 8



Permutasi dan Notasi siklis pada kode *Dandelion*. *Tree* dapat dibentuk melalui kode sederhana dari kode *Dandelion*. Langkah-langkah membentuk *tree* dari sebuah *Dandelion Code* yaitu:

1. Menuliskan kode dalam bentuk permutasi.
2. Menentukan semua notasi *siklis* yang terdapat dalam permutasi.
3. Menuliskan notasi *siklis* sebagai *path* dari *tree* yang akan dikonstruksi.
4. Mengkonstruksi *tree* graf sementara melalui $path(p_i, p_{i+1}, \dots, p_n)$ yang diperoleh pada langkah
(3) yaitu $1 \rightarrow p_i \rightarrow p_{i+1} \rightarrow \dots \rightarrow p_n$.
5. Mengkonstruksi *tree* graf melalui penggabungan *tree* sementara pada langkah (4) dan semua edge yang dibentuk berdasarkan permutasi.

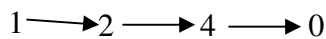
Contoh 1.

Diberikan kode *Dandelion*: (4,2,2,6,4), permutasi $\begin{pmatrix} 2 & 4 \\ 4 & 2 \end{pmatrix} \rightarrow$

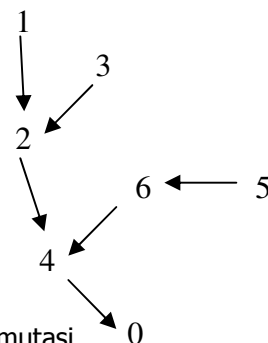
Notasi *Siklis* yang didapat dibentuk adalah: (2,4)

Path(1): (2,4)

Mengkonstruksi *tree* graf sementara:



Konstruksi *tree* graf dari code (4,2,2,6,4):



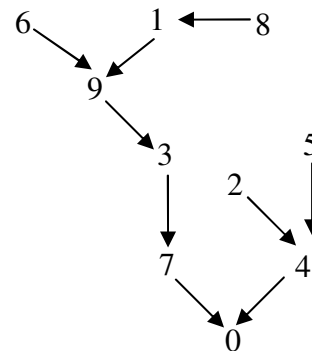
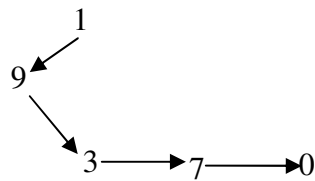
Contoh 2.

Diberikan kode *Dandelion*: (4,7,0,4,9,3,1,9), permutasi $\begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 7 & 0 & 4 & 9 & 3 & 1 & 9 \end{pmatrix}$

Notasi *siklis* yang dapat dibentuk adalah: $(9)(3,7)$

Path dari notasi *siklis* adalah: $(9,3,7)$ Mengkonstruksi tree graf sementara:

Konstruksi graf dari kode $(4,7,0,4,9,3,1,9)$



KESIMPULAN

Teknik mengkonstruksi tree dari suatu graf berarah dengan kode dandelion dapat dilakukan dengan menggunakan *spanning tree* dari graf ataupun dengan permutasi siklis. Pada proses pengkodean, pada akhir pembentukan kode, *tree* yang dikodekan akan membentuk suatu *star* dimana semua vertex mengarah ke *vertex 1*, dan dari *vertex 1* mengarah ke *vertex 0*.

DAFTAR PUSTAKA

Ahuja, R.K., T.L Magnanti and J.B. Orlin (1993), *Network Flow: Theory, Algorithms and Applications*, Chapter 13, Prentice Hall, Englewood Cliff, New Jersey, USA.

Caccetta, L. and Vijayan, K. (1987), 'Applications of Graph theory', *Ars Combinatoria*, vol. **23** (B), pp 21-77.

Nandang. 2004. "Encode" *Kamus Komputer & Teknologi Informas* ihttp://www.total.or.id/info.php?kk=encode. 26 Oktober 2009. Pkl.14.05.WIB.

Picciotto, S. 1999. *How to Encode a Tree*. <http://www.piccotto.org/sally/thesis.pdf>. 20 Agustus 2009. Pkl. 15:00.WIB.

