PENGELOMPOKAN DOKUMEN MENGGUNAKAN ALGORITMA DIG (*DOCUMENT INDEX GRAPH*)

Shofi Nur Fathiya (13508084)

Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung
e-mail: if18084@students.if.itb.ac.id

ABSTRAK

Berkembangnya teknologi di dunia menyebabkan arus informasi berjalan lebih mudah. Informasi bisa didapatkan dari mana saja dan kapan saja. Bentuknya pun beragam, dapat berupa tulisan ataupun lisan. Biasanya, informasi berupa tulisan disajikan dalam bentuk dokumen. Banyaknya dokumen yang ada seringkali membuat orang kebingungan, apakah dokumen yang satu dan lainnya berada dalam satu lingkup bahasan yang sama atau tidak. Untuk menyelesaikan masalah tersebut, digunakan metode *document clustering*, yang merupakan metode pengelompokan dokumen dengan cara mencocokkan dokumen satu dengan dokumen lainnya. Bila terdapat banyak kesamaan antara dokumen-dokumen tersebut, maka dokumen-dokumen tersebut dapat dikatakan satu kelompok. Metode *document clustering* memiliki beberapa cara penyelesaian, salah satunya adalah dengan menggunakan algoritma DIG (*Document Index Graph*). Sesuai dengan namanya, algoritma DIG ini menggunakan representasi graf dalam pengelompokan dokumen. Algoritma ini memperhatikan setiap kata dan frasa yang ada pada dokumen kemudian membandingkannya. Setelah melakukan perbandingan, maka akan terlihat apakah dokumen-dokumen tersebut memiliki keterkaitan dan berada pada bahasan yang sama atau tidak.

Kata kunci: Graf, *document clustering*, algoritma DIG.

1. PENDAHULUAN

Seiring dengan berkembangnya teknologi, semakin mudah orang mendapatkan informasi. Informasi yang didapat bermacam-macam, dapat berupa lisan maupun tulisan. Biasanya, informasi berupa tulisan disajikan dalam bentuk teks atau dokumen. Cara untuk mendapatkan informasi dalam bentuk dokumen ini sangatlah mudah, dapat diperoleh dari mana saja dan kapan saja, terutama setelah hadirnya internet dalam kehidupan kita. Ini menyebabkan dokumen-dokumen dapat diperoleh dengan jumlah yang sangat banyak dalam waktu yang singkat.

Banyaknya dokumen ini menyebabkan banyak orang bingung, apakah antara satu dokumen dengan dokumen lainnya memiliki keterkaitan atau tidak, dan juga apakah dua buah dokumen atau lebih berada dalam satu pokok bahasan yang sama ataukah tidak. Hal ini sangat perlu untuk diketahui agar dokumen tersebut tidak salah diartikan. Salah satu cara untuk mengetahui sejauh mana keterkaitan antar dokumen dan apakah berada pada bahasan yang sama atau tidak, digunakan metode *document clustering.*

Document clustering merupakan salah satu metode untuk mencari tahu keterkaitan antar dokumen dengan cara mengelompokkan dokumen-dokumen tersebut. Pengelompokan dilakukan berdasarkan kata dan frasa yang ada pada setiap dokumen.

Ada beberapa cara yang dapat digunakan dalam metode *document clustering*, beberapa diantaranya adalah *Suffix Tree, Single Pass Clustering, K-Nearest Neighbour,* dan algoritma *Document Index Graph.* Namun, dalam makalah ini hanya satu cara yang dibahas, yaitu Algoritma DIG (*Document Index Graph*). Sesuai dengan namanya, algoritma ini memakai representasi graf dalam pengelompokan dokumen. Graf yang dibangun merupakan graf berarah dimana arah tersebut menunjukkan struktur kalimat.

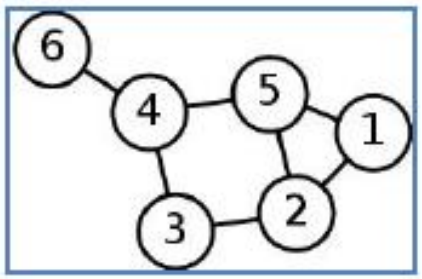
2. DASAR TEORI

Graf adalah sekumpulan benda-benda yang disebut simpul (*node / vertex*) yang dihubungkan oleh sisi (*edge*). Biasanya graf digambarkan sebagai kumpulan titik-titik (sebagai simpul) yang dihubungkan oleh garis-garis (sebagai sisi). Suatu graf G dapat dinyatakan sebagai G = <V,E> dimana V merupakan himpunan dari simpul yang berada pada G dan E merupakan himpunan sisi pada G. [4]

Selain simpul dan sisi, terdapat pula lintasan (*path*). Lintasan merupakan jalur yang harus ditempuh untuk mencapai suatu simpul dari simpul lain. Panjang lintasan

merupakan banyaknya sisi yang terdapat pada lintasan tersebut.[1]

Berikut adalah salah satu contoh dari graf:

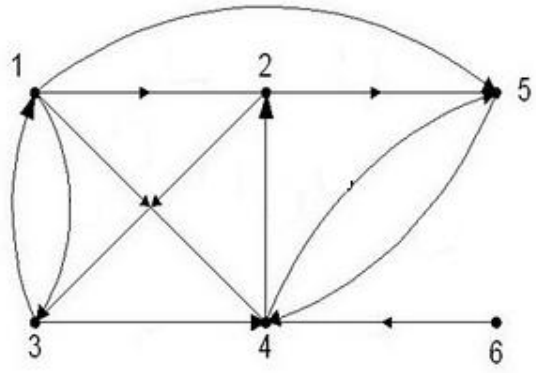


Gambar 1. Contoh graf

Dari gambar graf di atas, dapat dilihat bahwa:

1. Simpul pada graf dituliskan sebagai V = {1,2,3,4,5,6}
2. Sisi pada graf dituliskan sebagai E = {(1,2),(1,5),(2,3),(2,5),(3,4),(4,5),(4,6)}
3. Salah satu lintasan untuk mencapai simpul 3 dari simpul 1 adalah 1-2-3 dimana panjang lintasannya adalah 2.

Berdasarkan orientasi arah pada sisi, secara umum graf dapat dibagi menjadi 2 jenis, yaitu graf berarah dan graf tak berarah. Graf berarah merupakan graf dimana setiap sisinya memiliki arah sehingga simpul (1,2) dan (2,1) tidaklah sama. Sebaliknya, pada graf tak berarah, simpul (1,2) dan (2,1) dianggap sama. Gambar 1 di atas merupakan contoh graf tak berarah. Sedangkan contoh dari graf berarah adalah sebagai berikut.



Gambar 2. Graf berarah

Simpul yang terdapat pada graf berarah di atas adalah V = {1,2,3,4,5,6} dan sisinya adalah E = {(1,2) , (1,3) , (1,4) , (1,5) , (2,3) , (2,5) , (3,1) , (3,4) , (4,2) , (4,5) , (5,4) , (6,4)}. Dapat dilihat dari gambar bahwa sisi (1,3) dan (3,1) tidaklah sama karena keduanya memiliki asal dan arah simpul yang berbeda. Karena sisinya berbeda, maka lintasan yang dihasilkan pun berbeda. Bila graf di atas bukan merupakan graf berarah, maka untuk mencapai simpul 6 dari simpul 1 dapat melewati simpul 4, sehingga lintasan yang dibentu adalah 1-4-6. Namun karena graf tersebut merupakan graf, maka tidak ada lintasan yang dapat dilalui untuk mencapai simpul 6 dari simpul 1. Ini disebabkan tidak ada satu sisi pun yang mengarah ke simpul 6.

3.APLIKASI GRAF DALAM PENGELOMPOKAN DOKUMEN

Dalam penggunaan algoritma DIG, graf yang dibangun adalah graf berarah. Dalam graf ini, arah dari setiap sisi menunjukkan struktur kalimat yang ada pada setiap dokumen. Graf ini dibangun dari :

1. Simpul
   Simpul merupakan kata unik yang ada pada setiap dokumen. Setiap kata yang terdapat pada dua buah dokumen yang sedang dibandingkan harus terdapat pada himpunan simpul graf.
2. Sisi
   Sisi merupakan penghubung antarsimpul. Pada sisi terdapat informasi berupa nomor sisi yang menunjukkan posisi kata dalam kalimat dan dalam dokumen. Karena graf ini merupakan graf berarah, maka sisi dalam graf ini pun memiliki arah. Arah yang ditunjukkan menunjukkan urutan kata pada dokumen.
3. Lintasan
   Lintasan yang dibentuk dari simpul dan sisi merupakan representasi sebuah kalimat tertentu. Pada algoritma DIG, setiap kalimat pada setiap dokumen akan diproses satu per satu. Setiap kata yang belum ada di dalam kumpulan graf akan ditambahkan sebagai simpul. Sedangkan jika kata tersebut sudah ada dalam kumpulan graf, maka akan ditambahkan sisi baru.
   Untuk setiap kata yang bertetangga dihubungkan dengan sisi. Untuk mendapatkan *matching phrase*, dibuatkan daftar data dokumen-dokumen yang mempunyai sisi serupa ke dalam sebuah tabel. Jika *matching phrase* berikutnya mempunyai sisi yang merupakan kelanjutan dari sisi sebelumnya, maka *matching phrase* tersebut digabungkan dengan *matching phrase* sebelumnya. Begitu seterusnya sampai seluruh dokumen selesai diproses.[5]

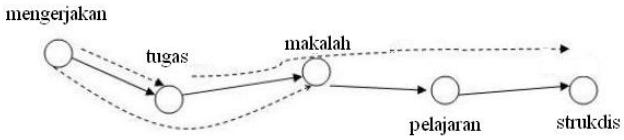
Berikut adalah ilustrasi pembentukan graf menggunakan algoritma DIG. Pada ilustrasi ini hanya digunakan tiga buah dokumen saja, yaitu dokumen A, dokumen B, dan dokumen C, dimana masing-masing dokumen memiliki kalimat yang terdiri dari dua atau lebih kata. Ilustrasinya adalah sebagai berikut:

Dokumen A
Pada dokumen A, terdapat kalimat-kalimat:
mengerjakan tugas
mengerjakan tugas makalah
tugas makalah pelajaran strukdis

Dokumen B :
Pada dokumen B terdapat kalimat-kalimat:
tugas laporan
tugas makalah wajib

Dokumen C :
Pada dokumen C terdapat kalimat-kalimat:
laporan praktikum
praktikum kimia
anggota kelompok praktikum
pelajaran kimia

Langkah pertama yang dilakukan adalah membuat graf berdasarkan kalimat-kalimat pada dokumen A dimana setiap kata yang terdapat pada dokumen A menjadi simpul dalam graf.
Graf yang dibentuk dari dokumen A adalah :

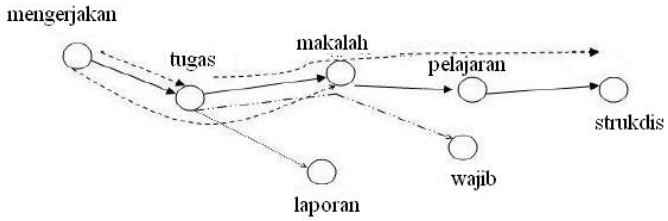


Gambar 3. Graf A yang dibentuk dari dokumen A

Dari gambar graf di atas, terlihat bahwa graf tersebut memiliki:

1. Simpul
   Simpul yang terdapat pada Graf A adalah seluruh kata yang terdapat pada dokumen A, yaitu V = {mengerjakan,tugas,makalah,pelajaran,strukdis}.
2. Sisi
   Sisi yang terdapat pada Graf A adalah kata-kata yang letaknya bersebelahan pada dokumen A, yaitu E = {(mengerjakan,tugas) , (tugas,makalah) , (makalah,pelajaran) , (pelajaran,strukdis)}
3. Lintasan
   Lintasan yang terbentuk pada Graf A berjumlah 3 buah, yaitu :
   a. mengerjakan – tugas
   b. mengerjakan – tugas – makalah
   c. tugas – makalah – pelajaran – strukdis

Setelah mendapatkan Graf A, maka langkah selanjutnya adalah membuat graf yang meliputi dokumen A dan dokumen B, yaitu Graf B. Penggambaran dari Graf B adalah sebagai berikut:



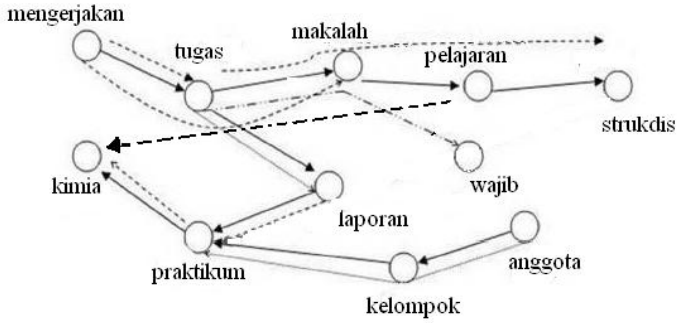
Gambar 4. Graf B yang dibentuk dari dokumen A dan dokumen B

Dari gambar graf B di atas, terlihat bahwa graf tersebut memiliki:

1. Simpul
   Simpul yang terdapat pada Graf B adalah seluruh kata yang terdapat pada dokuman A dan dokumen B, yaitu V = {mengerjakan,tugas,makalah, pelajaran,strukdis, laporan,wajib}
2. Sisi
   Sisi yang terdapat pada Graf B adalah E = {(mengerjakan,tugas) , (tugas,makalah) , (makalah,pelajaran) , (pelajaran,strukdis) , (tugas laporan) , (makalah wajib)}
3. Lintasan
   Lintasan yang terbentuk pada Graf A berjumlah 3 buah, yaitu :
   a. mengerjakan – tugas
   b. mengerjakan – tugas – makalah
   c. tugas – makalah – pelajaran – strukdis
   d. tugas – laporan
   e. tugas – makalah - wajib

Dari gambar graf B pula, terlihat adanya keterkaitan antara dokumen A dan dokumen B, yaitu pada sisi (tugas laporan) dan (makalah wajib).

Setelah mendapatkan Graf A dan Graf B, langkah selanjutnya adalah membuat graf yang meliputi dokumen A, dokumen B, dan dokumen C, yaitu Graf C. Penggambaran dari Graf C adalah sebagai berikut:



Gambar 5. Graf C yang dibentuk dari dokumen A, dokumen B, dan dokumen C

Dari gambar graf C di atas, terlihat bahwa graf tersebut memiliki:

1. Simpul
   Simpul yang terdapat pada Graf A adalah V = {mengerjakan,tugas,makalah,pelajaran,strukdis, laporan,wajib,praktikum,kimia,anggota,kelompok}
2. Sisi
   Sisi yang terdapat pada Graf A adalah E = {(mengerjakan,tugas) , (tugas,makalah) , (makalah,pelajaran) , (pelajaran,strukdis) , (tugas laporan) , (makalah wajib) , (laporan,praktikum) , (praktikum,kimia) , (anggota,kelompok) , (kelompok,praktikum) , (pelajaran kimia)}
3. Lintasan
   Lintasan yang terbentuk pada Graf A berjumlah 3 buah, yaitu :
   a. mengerjakan – tugas
   b. mengerjakan – tugas – makalah
   c. tugas – makalah – pelajaran – strukdis
   d. tugas – laporan
   e. tugas – makalah – wajib
   f. laporan – praktikum
   g. praktikum – kimia
   h. anggota – kelompok – praktikum
   i. pelajaran – kimia

Dari gambar graf C pula, terlihat adanya keterkaitan antara dokumen A, dokumen B dan dokumen C. Keterkaitan antara dokumen A dan dokumen B terdapat pada sisi (tugas laporan) dan (makalah wajib). Keterkaitan antara dokumen B dan dokumen C terdapat pada sisi (laporan,praktikum). Terakhir, keterkaitan antara dokumen A dan dokumen C terdapat pada sisi (pelajaran,kimia).

Contoh kasus di atas merupakan contoh kasus yang mencari keterkaitan antara 3 buah dokumen. Jumlah kata dan frasa dalam setiap dokumen pun hanya sedikit. Padahal dalam kenyataannya, jumlah dokumen yang ada di dunia ini sangatlah banyak, tidak sekadar 10 atau 100 buah. Dalam setiap dokumen pun jumlah kata dan frasanya bisa mencapai ribuan bahkan lebih. Akan sangat sulit dilakukan bila graf dibuat secara manual. Perlu ada penanganan yang lebih baik agar setiap dokumen yang ada dapat tergambarkan keterkaitannya. Untuk itulah dibuat algoritma DIG ini agar proses mengaitkan dokumen-dokumen dapat dilakukan dengan lebih cepat dan mudah.

Algoritma DIG memiliki struktur sebagai berikut.

DIG incremental construction and phrase matching
Require : G_{i-1} : cumulative graph up to document d_{i-1} or G_0 if no documents were processed previously

d_i ← Next Document
M ← Empty List {M is a list of matching phrases from previous documents}

for each sentences sij in di **do**
  v_1 ← t_{ij1} {first word in sij}
  if v_1 is not in G_{i-1} **then**
  Add v_1 to G_{i-1}
  end if
  for each term $t_{ijk} \in$ sij, k=2 ,….,tij **do**
    v_k ← t_{ijk}; v_{k-1}
    t_{ijk-1}) ; ← $e_k=(v_{k-1},v_k)$
    if vk is not in G_{i-1} **then**
      Add vk to G_{i-1}
    end if
    if e_k is an edge in G_{i-1} **then**
      Retrieve a list of document entries from v_{k-1} document table that have a sentence on the edge e_k
      Extend previous matching phrases in *M* for phrases that continue along edge e_k
    else
    Add edge e_k to G_{i-1}
    end if
    Update sentence path in nodes v_{k-1} and v_k
  end for
end for
G_i ← G_{i-1}
Output machine phrases list in M

Gambar 6. Algoritma DIG (*Document Index Graph*)

Terkadang, selain mencari tahu keterkaitan antar dokumen, kita juga ingin mengetahui seberapa besar hubungan antar dokumen. Hubungan antardokumen ini digambarkan dalam sutu nilai tertentu sehingga mudah ketika dibandingkan dengen hubungan antardokumen lain. Ada beberapa pendekatan yang dapat digunakan untuk mengetahui nilai dari keterkaitan antardokumen.[3]

1. *Single term*
   Single term (kesamaan dokumen berbasis kata) merupakan cara untuk menghitung nilai kesamaan antardokumen dilihat dari kata (*term*) yang ada pada masing-masing dokumen. Nilai kesamaan ini dapat diperoleh dengan metode *Cosine based Similarity*. Dengan mengukur dua vektor berdimensi n, dapat dicari sudut di antara keduanya untuk kemudian dicari nilai cosinus dari sudut tersebut.
   Untuk *text-matching*, atribut yang digunakan adalah vektor TF-IDF. Nilai kesamaan dokumen d1 dan d2 dihitung berdasarkan persamaan (1) berikut :

Persamaan (1) :

$$simt(d1,d2) = \cos(d1,d2) = \frac{d1\,d2}{|d1|\,|d2|}$$

d1 = dokumen 1
d2 = dokumen 2

TF atau term frequency merupakan banyaknya term dalam sebuah dokumen. Nilai TF dapat diperoleh dari persamaan:

Persamaan (2):

$$tf = \frac{f}{m} \qquad f > 0$$

f = frekuensi term dalam sebuah dokumen
m = frekuensi maksimum dari suatu term yang terdapat dalam sebuah dokumen

Sedangkan untuk IDF atau *Inverse Document Frequency* , yang merupakan banyaknya term tertentu dalam keseluruhan dokumen dapat dicari nilainya melalui persamaan:

Persamaan (3):

$$idf_j = \log_2\left(\frac{n}{n_j}\right) + 1 \qquad n_j > 0$$

n = jumlah seluruh dokumen
n_j = jumlah dokumen yang mempunyai term j

2. *Phrase based similarity*

Phrase based similarity merupakan pendekatan untuk menghitung kesamaan dokumen berdasarkan frasa yang dimiliki setiap dokumen. Dengan memperhatikan urutan kata yang terdapat pada kedua dokumen yang sedang dibandingkan diharapkan akan meningkatkan nilai akurasi pengelompokan dokumen.

Nilai kesamaan dokumen didasarkan pada *shared phrase* pada masing-masing pasangan dokumen yang dibandingkan. *Shared phrase* merupakan frasa -- yang terdapat pada kedua dokumen. Faktor penetu *shared phrase* dalam menentukan kesamaan dokumen adalah:

a. jumlah *matching phrase*
b. panjang *matching phrase*
c. frekuensi *matching phrase* pada kedua dokumen
d. level signifikan (*weight*) dari *matching phrase* di kedua dokumen.

Rumus untuk menentukan nilai kesamaan antara dokumen d1 dan dokumen d2 berdasarkan frasa adalah:

Persamaan (4):

$$simp(d_1, d_2) = \frac{\sqrt{\sum_{i=1}^{n}[g(1i).(f1i\,w1i+f2i\,w2i)]^2}}{\sum_j |s1j| \cdot w1j + \sum_k s2k| \cdot w2k}$$

$$g(1i) = (1i/|si|)^y$$

3. Gabungan antara *single term* dan *phrase based similarity*

Kesamaan dokumen akhir dihitung dari kombinasi antara kesamaan berbasis kata dan kesamaan frasa dengan rumus berikut.

Persamaan (5):

$$sim(d_1, d_2) = \alpha \cdot simp(d_1, d_2) + (1-\alpha) \cdot simt(d_1, d_2)$$

Setelah mendapatkan nilai kesamaan dari dua buah dokumen, maka dapat dilihat sejauh apakah kedua dokumen tersebut berkaitan.

4. KESIMPULAN

Mengetahui keterkaitan antardokumen merupakan suatu hal yang perlu dilakukan saat ini, mengingat banyaknya dokumen dan informasi yang terkandung di dalamnya. Algoritma DIG (*Document Index Graph*) merupakan salah satu cara yang dapat digunakan untuk mengetahui keterkaitan antara dokumen satu dengan dokumen lainnya. Algoritma ini termasuk dalam metode *clustering document*, yaitu metode yang menerapkan pengelompokan dokumen berdasarkan kata dan frasa yang terdapat pada setiap dokumen.

REFERENSI

[1] Murni, Rinaldi, "Matematika Diskrit edisi Ketiga", Informatika, 2005.

[2] Huang, Ronghuai, "Advanced Data Mining and Applications", Springer, 2009.

[3] Ernawati, Sari, dkk, "Klusterisasi Dokumen Berita Berbahasa Indonesia Menggunakan Document Index Graph", 2009, hal 2-3.

[4] http://id.wikipedia.org/wiki/Teori_graf
Tanggal akses : 19 Desember 2009

[5] http://www.ittelkom.ac.id/library/index.php?view=article&catid=20%3Ainformatika&id=563%3Aalgoritma-document-index-graph-dig&option=com_content&Itemid=15
Tanggal akses : 19 Desember 2009