# WAND Top-K Retrieval

Alfan F. Wicaksono Fakultas Ilmu Komputer, Universitas Indonesia

Berdasarkan paper "Efficient Query Evaluation using a Two-Level Retrieval Process", by Andrei Z. Broder, D. Carmel, M. Herscovici, A. Soffer, J. Zien



Ada 22 Milyar hasil yang relevan, namun yang ditampilkan hanya sekitar Top-10 saja.







SafeSearch aktif

to be or not to be Q Semua Gamba ▶ Video Buku Berita ! Lainnva Alat Sekitar 22.810.000.000 hasil (0.55 detik) Kiat: Telusuri hasil dalam bahasa Indonesia saja. Anda dapat menentukan bahasa penelusuran di Preferensi "To be, or not to be" is the opening phrase of a soliloguy given by Prince Hamlet in the so-called "nunnery scene" of William Shakespeare's play Hamlet, Act 3, Scene 1. In the speech, Hamlet contemplates death and suicide, bemoaning the pain and unfairness of life but acknowledging that the alternative might be worse. https://en.wikipedia.org > wiki > To be, or not to be To be, or not to be - Wikipedia Tentang cuplikan pilihan • III Masukan Orang juga bertanya What is the full quote To be, or not to be?  $\vee$ Why does Hamlet say To be, or not to be?

What does Shakespeare say To be, or not to be?

But Space 
Control Space 
Service and Space 
S

o be, or not to be—that is the question:
hether 'tis nobler in the mind to suffer
be slings and arrows of outrageous fortune
r to take arms against a sea of troubles
nd by opposing end them. To die, to sleepo more—and by a sleep to say we end
he heartache, and the thousand natural shock
hat flesh is heir to. 'Tis a consummation
evoutly to be wished. To die, to sleepo sleep--perchance to dream: ay, there's the ru
wr in that sleep of death what dreams may con
hen we have shuffled off this mortal coil,
lust give us pause.

om. To be, or not to be, I there's the point, Die, to fleepe, is that all? I all: o fleepe, to dreame, I mary there it goes, n that dreame of death, when wee awake, borne before an euerlafting Iudge, n whence no paffenger euer retur nd, vndifcouered country, at whose fight happy smile, and the accorded damn'd, for this, the ioyfull hope of this, o'd beare the scornes and flattery of the worned by the right rich, the rich curffed of the p

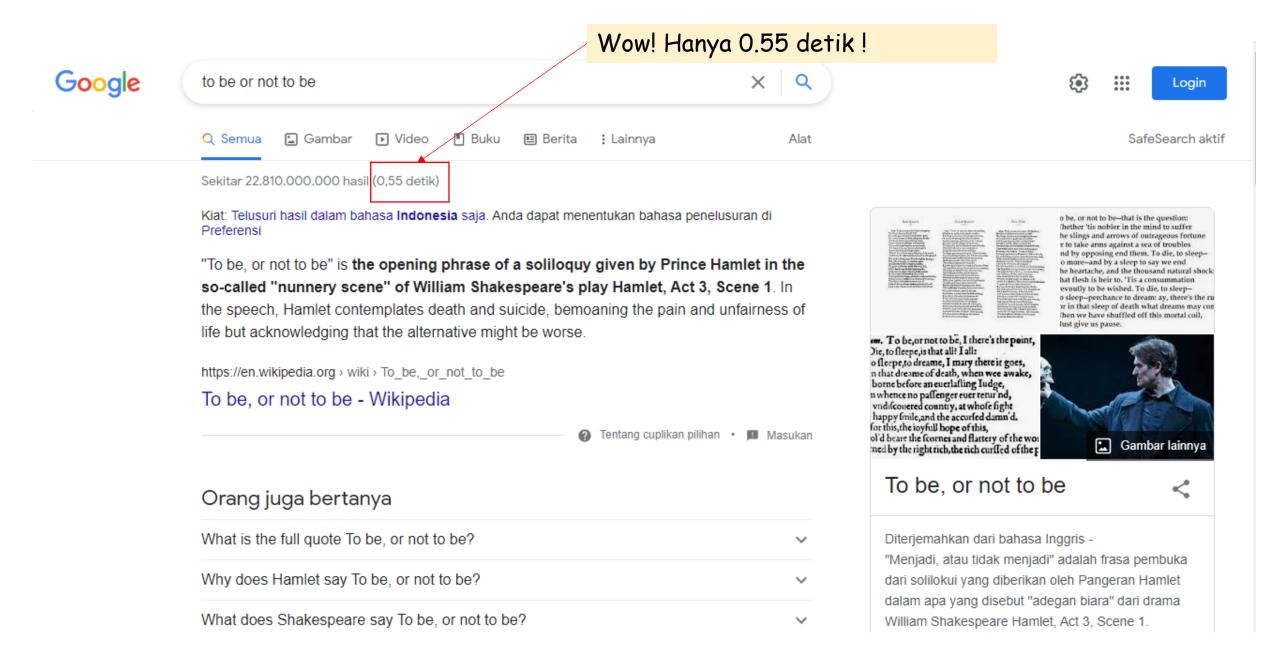


#### To be, or not to be



Diterjemahkan dari bahasa Inggris -

"Menjadi, atau tidak menjadi" adalah frasa pembuka dari solilokui yang diberikan oleh Pangeran Hamlet dalam apa yang disebut "adegan biara" dari drama William Shakespeare Hamlet, Act 3, Scene 1.



## Scoring Method

$$score(Q,D) = \sum_{t \in Q \cap D} \alpha_t w(t,D)$$

Bagian yang tidak bergantung pada dokumen (bergantung pada query) Bagian yang bergantung pada dokumen D

## Scoring Method

$$score(Q, D) = \sum_{t \in Q \cap D} \alpha_t w(t, D)$$

Contoh: TF-IDF

$$\alpha_t = tf(t,Q) \cdot log(N/df(t))$$
; boleh juga dengan  $\alpha_t = log(N/df(t))$ 

$$w(t,D) = tf(t,D)$$
 ; boleh juga dengan  $w(t,D) = 1 + \log(tf(t,D))$ 

Juga dengan varian TF-IDF yang lainnya

## Scoring Method

$$score(Q, D) = \sum_{t \in Q \cap D} \alpha_t w(t, D)$$

Contoh: BM25

$$\alpha_t = \log(N/df(t)) \qquad w(t,D) = \frac{(k_1+1).tf(t,D)}{k_1\left((1-b) + b\frac{dl}{avdl}\right) + tf(t,D)}$$

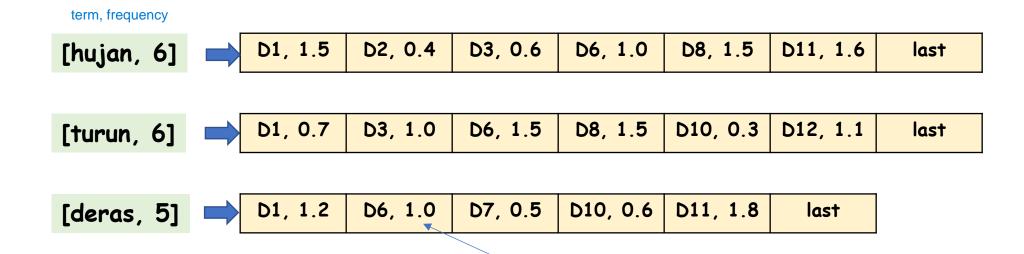
Juga dengan varian BM25 yang lainnya

### Term-at-a-Time (TaaT)

```
SCORE(q):
   float Scores[N] = 0
   for each query term t do
      calculate \mathbf{w}(t,q) and fetch postings list for t
      for each pair(d, tf(t,d)) in postings list do
5
          Scores[d] += \mathbf{w}(t,d) \times \mathbf{w}(t,q)
6
   return Top K components of Scores[]
```

Term at a time

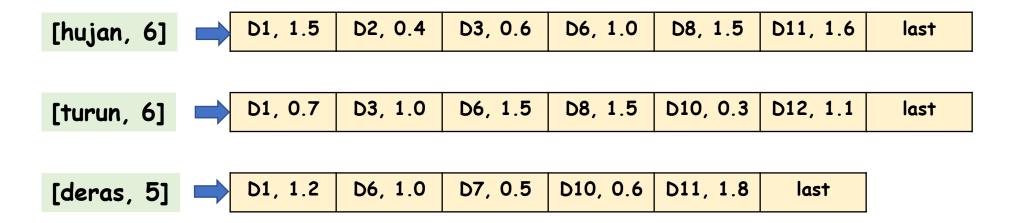
yang di simpen bukan score w(t, d) tapi gak disimpen di inverted index karena bnetuknya floating point number.



Misal, score w(t,d) sudah dihitung (bisa dengan TF-IDF, BM25 atau yang lainnya). Ini hanya untuk kemudahan saat visualisasi. Kenyataannya, nilai ini bukan sesuatu yang pre-computed dan disimpan di postings lists.

Yang pre-computed dan simpan biasanya adalah **tf(t,d)** yang berupa integer.

### TaaT



Scores = []

Scores = [(D1, 1.5)]

Scores = [(D1, 1.5), (D2, 0.4), ]

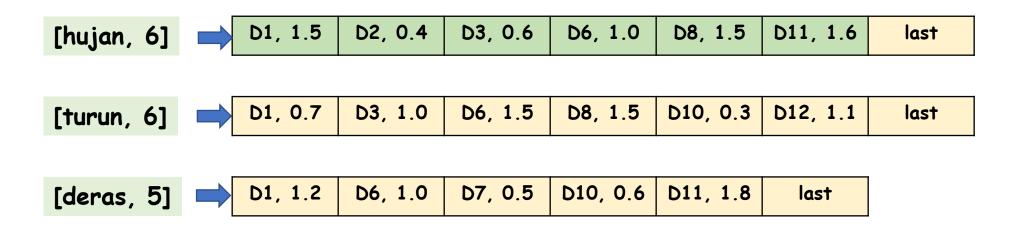
Scores = [(D1, 1.5), (D2, 0.4), (D3, 0.6)]

Scores = [(D1, 1.5), (D2, 0.4), (D3, 0.6), (D6, 1.0)]

### TaaT

Scores = [(D1, 1.0), (D2, 0.4), (D3, 0.6), (D6, 1.0), (D8, 1.5)]

### TaaT



Scores = [(D1, 1.5), (D2, 0.4), (D3, 0.6), (D6, 1.0), (D8, 1.5), (D11, 1.6)]

### TaaT

Scores = [(D1, 2.2), (D2, 0.4), (D3, 0.6), (D6, 1.0), (D8, 1.5), (D11, 1.6)]

Scores = [(D1, 2.2), (D2, 0.4), (D3, 1.6), (D6, 1.0), (D8, 1.5), (D11, 1.6)]

Dan, seterusnya, silakan lanjutkan sendiri ...
Jika score semua dokumen sudah dihitung, gunakan HEAP untuk cari Top-K

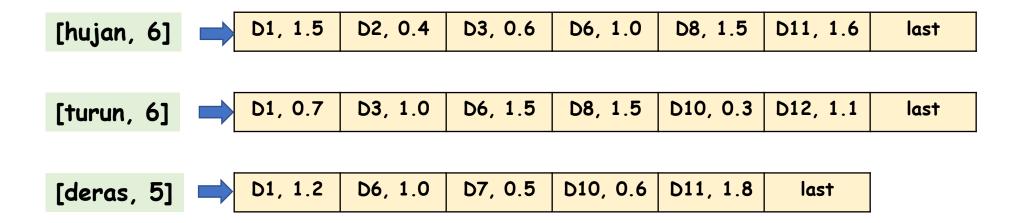
### Document-at-a-Time (DaaT)

 Semua postings lists yang terkait query diproses secara paralel.

 Score dari sebuah dokumen dihitung secara penuh (fully evaluated) sebelum berpindah ke dokumen yang lain.

• Di setiap list ada pointer ke current Doc ID. Selalu majukan pointer pada list dengan currect Doc ID paling kecil.

### DaaT



```
Scores = []
```

DaaT

Jumlah dokumen yang fully-evaluated: 1

3.4 itu dapet dari 1.5 + 0.7 + 1.2 (hal ini karena D1 semuanya)

Scores = 
$$[(D1, 3.4), ]$$

DaaT

Jumlah dokumen yang fully-evaluated: 2

Kalau misal ada D2, D3, D6, kita pilih yang paling kecil index dokumennya nya. Nah baru kita masukkin (pointer yang D2 dimajuin)

Scores = 
$$[(D1, 3.4), (D2, 0.4)]$$

DaaT

Jumlah dokumen yang fully-evaluated: 3

Sekarang yang paling kecil itu D3, nah kita ambil 0.6 + 1, terus majuin lagi pointernya

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6)]

DaaT

Jumlah dokumen yang fully-evaluated: 4

3 3 nya pointenrya majuin

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6), (D6, 3.5)]

DaaT

Jumlah dokumen yang fully-evaluated: 5

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6), (D6, 3.5), (D7, 0.5)]

DaaT

Jumlah dokumen yang fully-evaluated: 6

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6), (D6, 3.5), (D7, 0.5), (D8, 3.0)]

DaaT

Jumlah dokumen yang fully-evaluated: 7

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6), (D6, 3.5), (D7, 0.5), (D8, 3.0), (D10, 0.9)]

DaaT

Jumlah dokumen yang fully-evaluated: 8

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6), (D6, 3.5), (D7, 0.5), (D8, 3.0), (D10, 0.9), (D11, 3.4)]

DaaT

Jumlah dokumen yang fully-evaluated: 9

Scores = [(D1, 3.4), (D2, 0.4), (D3, 1.6), (D6, 3.5), (D7, 0.5), (D8, 3.0), (D10, 0.9), (D11, 3.4), (D12, 1.1)]

DaaT

Jumlah dokumen yang fully-evaluated: 9

## Masalah dengan DaaT sebelumnya

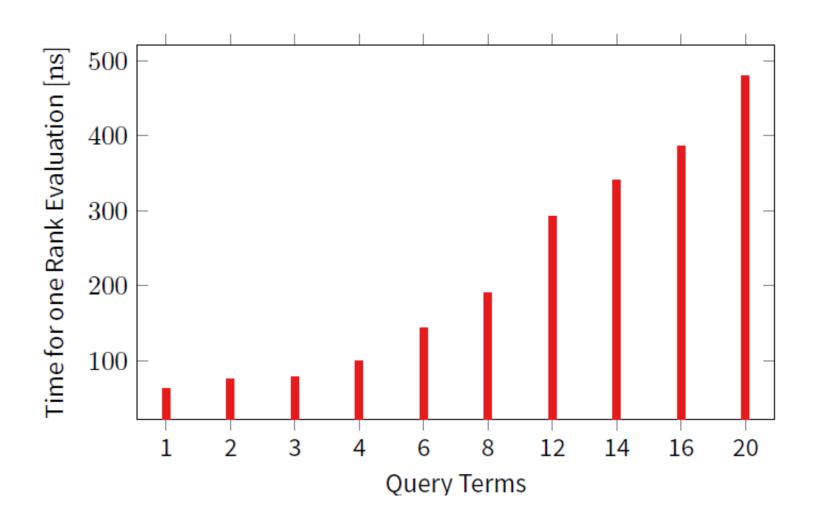
Best casenya itu O(jumlah dokumen)

- Semua dokumen harus dihitung score-nya (fully-evaluated)
- Menghitung score dari sebuah dokumen (misal dengan BM25) sangat mahal! (lihat slide berikutnya)

Sangat mahal hitung score tapi kalo querynya term nya makin banyak maka makin mahal

 Bisakah kita mencari Top-K documents tanpa harus menghitung score dari semua dokumen pada postings lists terkait?

## BM25 Computation Speed



## BM25 Computation Speed

 Menghitung BM25 untuk sebuah pasangan query dan dokumen memerlukan kira-kira 100 nanoseconds.

 Misal, query "the student" cocok dengan 25M dokumen. Kita tahu bahwa "the" bisa muncul di hampir setiap dokumen.

 $25.000.000 \times 100$  nanoseconds = 2.5 seconds

Apakah user mau menunggu 2.5 detik untuk dapat jawaban?

### A/B Testing Experiment by Google

- Average revenue per user turun 4% ketika presentasi hasil search diperlambat.
- User sadar atau sensitive terhadap search system yang diperlambat minimal 50 ms.

### Bing Search Engine

- Mengurangi waktu proses sebesar 100 ms meningkatkan annual revenue sebesar 0.6%.
- Di tahun 2015, 1% kenaikan revenue per user setara dengan kenaikan sekitar \$10.000.000 per tahun.

## Strategi

#### Tambah server (horizontal scaling)

 Server bertambah -> perlu usaha maintenance lebih, system makin kompleks, power cost yang makin naik.

#### Ganti spesifikasi hardware yang lebih canggih (vertical scaling)

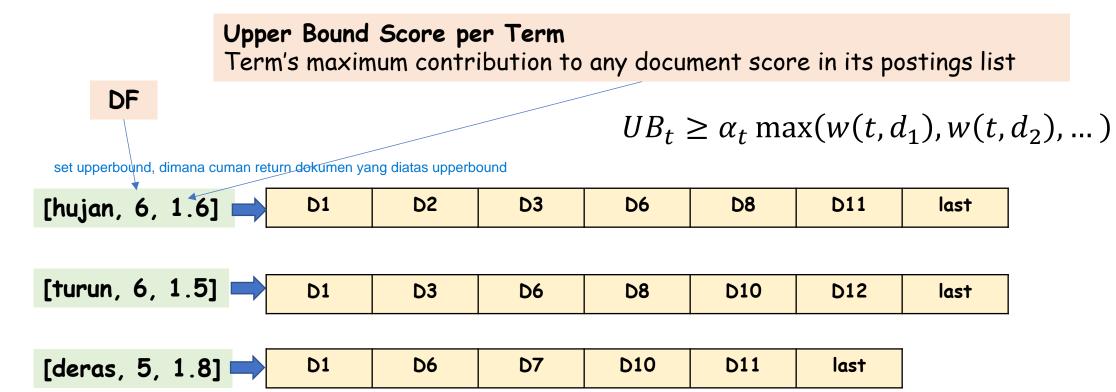
Harga mahal & tidak scaling terus-menerus (diminishing returns)

#### Perbaikan masih bisa di level Software/Program

· Usulkan algoritma query processing yang lebih cepat atau efisien

## Solusi: Top-K Retrieval

- Secara umum, search engine memang tidak menampilkan semua dokumen yang relevan; biasanya hanya Top-K saja (dengan K biasanya 10).
- For a given query, it's possible to produce a "complete" ranking without scoring all documents in the postings lists.
- Avoid scoring documents that we know will not appear in the Top-K results.



# WAND Operator

Digunakan untuk menentukan apakah sebuah dokumen punya kemungkinan untuk berada di Top-K.

$$WAND(x_1, UB_1, x_2, UB_2, x_3, UB_3, \dots, x_k, UB_k, \theta)$$

The minimum score in the current Top-K documents

= True jika 
$$UB(d,q) = \sum_{1 \le i \le k} x_i UB_i \ge \theta$$

x<sub>i</sub> bernilai 1 jika term i ada di dokumen; dan 0 jika sebaliknya.

di skip kalau dokumen gak mungkin melawan upperboud

Jika **True**, artinya dokumen tersebut punya chance untuk berada di Top-K. Jika **False**, sudah tidak ada chance. **Jangan dievaluasi! Skip saja!** 

## Two Level Approach

### Tahap 1 (WAND Condition)

- Cari kumpulan kandidat dokumen dengan menghitung UB(d,q)
- Jika UB(d,q) >= minimum score, ke tahap 2
- Jika tidak, di-skip tidak perlu hitung dokumen yang bm25 nya kurang dari minimum score

#### Tahap 2 (Full Evaluation)

- Hitung exact full score dari para kandidat yang lolos tahap 1
- Periksa apakah bisa mengalahkan minimum score

### Tahap Akhir (Top-K Heap)

TOP-1 :

Threshold:

Pivot posisi tengah (ada di sebelah kiri/sebelah kanan). Jadi ini buat nentuin posisi mana dokumen yang tidak layak/tidak layak masuk top K

curboc : sebelum pivot tidak mungkin top K

urut cari pivot terus ada if yang dilalui, ikutin tahap bawah lagi

1.6 dari D11 itu upperbound, dihitung secara 1 off (jadi cuman sekali ktia evaluasi (yaiut pas training) 6 = DF (document frequency)

[hujan, 6, 1.6] D1, 1.5 D2, 0.4 D3, 0.6 D6, 1.0 D8, 1.5 D11, 1.6 last

[turun, 6, 1.5] D1, 0.7 D3, 1.0 D6, 1.5 D8, 1.5 D10, 0.3 D12, 1.1 last

[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last

untuk setiap ngitung dokumen pakai bm25, komputasinya lama

kita ingin untuk membuang dokumen yang pasti engga masuk ke top K

jadi kita cuman itung dokumen yang punya

1. **Function** init(queryTerms) 2. terms  $\leftarrow$  queryTerms

Pivot : 0 curDoc : 0

: []

: 0

kesempatan jadi top k, lalu dihitung bm25 nya 2 .

 $2. \quad \text{terms} \leftarrow \text{query rem}$ 

3.  $\operatorname{curDoc} \leftarrow 0$ 

4. for each  $t \in \text{terms}$ 

5.  $posting[t] \leftarrow t.iterator.next(0)$ 

#### Inisialisasi

TOP-1

Threshold

- Sets the current document (curDoc) to be considered to zero
- For each query term, t, initializes its current posting to be the first posting item in the list

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
5.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
6.
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
8.
        pivot \leftarrow posting[pTerm].DID
9.
        if (pivot = lastID) return (NoMoreDocs)
10.
        if (pivot \leq curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
16.
              /* Success, all terms preceding pTerm belong
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
              /* not enough mass yet on pivot, advance
20.
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
1. Function next(\theta)
                                                                                                             2.
                                                                                                                 repeat
 TOP-1
                                                                                                                    /* Sort the terms in non-decreasing order of
                                                                                                                    DID */
 Threshold
                                                                                                                    sort(terms, posting)
                                                                                                             4.
                                                                                                                    /* Find pivot term - the first one with accumulated
                                                                                                             5.
 Pivot
                           : 0
                                                                                                                     UB > \theta */
                                                                                                                    pTerm \leftarrow findPivotTerm(terms, \theta)
 curDoc
                           : 0
                                       #documents that are fully evaluated: 0
                                                                                                                    if (pTerm = null) return (NoMoreDocs)
                                                                                                             7.
                                                                                                                    pivot \leftarrow posting[pTerm].DID
                                                                                                                    if (pivot = lastID) return (NoMoreDocs)
                                                                                                             9.
                                                                                                             10.
                                                                                                                    if (pivot < curDoc)
                                                                                                                      /* pivot has already been considered, advance
                                                                                                             11.
                                                                                                                       one of the preceding terms */
                                                                                                             12.
                                                                                                                      aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                                      posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                             13.
                                                                                                                    else /* pivot > curDoc */
                                                                                                             14.
                                                                                                             15.
                                                                                                                      if (posting[0].DID = pivot)
                                                                                                                         /* Success, all terms preceding pTerm belong
                                                                                                             16.
                                                                                                                         to the pivot */
pivot term itu selalu lihat dari query
                                                                                                             17.
                                                                                                                         curDoc \leftarrow pivot
pivot itu pointer yang ada di pivot term
                                                                                                             18.
                                                                                                                         return (curDoc, posting)
                                                                                                             19.
                                                                                                                      else
threashold di update dari maximum dokumen
                                                                                                             20.
                                                                                                                         /* not enough mass yet on pivot, advance
                                                                                                                         one of the preceding terms */
                                           Sudah terurut berdasarkan Doc ID
                                                                                                             21.
                                                                                                                         aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                             22.
                                                                                                                         posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                   udah terurut
                                                                                                             23.
                                                                                                                    end repeat
 [hujan, 6, 1.6]
                                 D1, 1.5
                                               D2, 0.4
                                                             D3, 0.6
                                                                            D6, 1.0
                                                                                          D8, 1.5
                                                                                                        D11, 1.6
                                                                                                                         last
                                 D1, 0.7
 [turun, 6, 1.5]
                                               D3, 1.0
                                                             D6, 1.5
                                                                            D8, 1.5
                                                                                         D10, 0.3
                                                                                                       D12, 1.1
                                                                                                                         last
                                               D6, 1.0
                                 D1, 1.2
                                                             D7, 0.5
                                                                           D10, 0.6
                                                                                         D11, 1.8
 [deras, 5, 1.8]
                                                                                                           last
```

```
1. Function next(\theta)
                                                                                                          2.
                                                                                                               repeat
TOP-1
                                                                                                                 /* Sort the terms in non decreasing order of
                                                                                                                  DID */
Threshold
                                                                                                                 sort(terms, posting)
                                                                                                                 / Find pivot term - the first one with accumulated
Pivot
                          : 0
                                                                                                                  UB > \theta */
                                                                                                          6.
                                                                                                                 pTerm \leftarrow findPivotTerm(terms, \theta)
curDoc
                          : 0
                                     #documents that are fully evaluated: 0
                                                                                                                 if (pTerm = null) return (NoMoreDocs)
                                                                                                          7.
                                                                                                                 pivot \leftarrow posting[pTerm].DID
                                                                                                                 if (pivot = lastID) return (NoMoreDocs)
                                                                                                          9.
                                                                                                                 if (pivot < curDoc)
                                                                                                          10.
                                                                                                                   /* pivot has already been considered, advance
                                                                                                          11.
                                                                                                                    one of the preceding terms */
                                                                                                          12.
                                                                                                                   aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                          13.
                                                                                                                   posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                                 else /* pivot > curDoc */
                                                                                                          14.
                                                                                                          15.
                                                                                                                   if (posting[0].DID = pivot)
                                                                                                          16.
                                                                                                                      /* Success, all terms preceding pTerm belong
                                                                                                                      to the pivot */
                                                                                                          17.
                                                                                                                      curDoc \leftarrow pivot
                                                                                                          18.
                                                                                                                      return (curDoc, posting)
                                                                                                          19.
                                                                                                                    else
                                                                                                          20.
                                                                                                                      /* not enough mass yet on pivot, advance
      pivotnya dapet yang pertama karena 1.6 > 0
                                                                                                                       one of the preceding terms */
                                         hujan adalah pivot term
                                                                                                                      aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                          21.
                                                                                                          22.
                                                                                                                      posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                          23.
                                                                                                                 end repeat
[hujan, 6, 1.6]
                               D1, 1.5
                                             D2, 0.4
                                                           D3, 0.6
                                                                          D6, 1.0
                                                                                       D8, 1.5
                                                                                                     D11, 1.6
                                                                                                                      last
                               D1, 0.7
[turun, 6, 1.5]
                                             D3, 1.0
                                                           D6, 1.5
                                                                          D8, 1.5
                                                                                       D10, 0.3
                                                                                                     D12, 1.1
                                                                                                                      last
[deras, 5, 1.8]
                               D1, 1.2
                                             D6, 1.0
                                                           D7, 0.5
                                                                         D10, 0.6
                                                                                       D11, 1.8
                                                                                                        last
```

```
1. Function next(\theta)
                                                                                                           2.
                                                                                                               repeat
TOP-1
                                                                                                                  /* Sort the terms in non decreasing order of
                                                                                                                  DID */
Threshold
                                                                                                                  sort(terms, posting)
                                                                                                                  /* Find pivot term - the first one with accumulated
Pivot
                                                                                                                   UB > \theta */
                                                                                                                  pTerm \leftarrow findPivotTerm(terms, \theta)
                                                                                                           6.
curDoc
                          : 0
                                      #documents that are fully evaluated: 0
                                                                                                                  if (nTorm - null) return (NoMore Doce)
                                                                                                           7.
                                                                                                                  pivot \leftarrow posting[pTerm].DID
                                                                                                           9.
                                                                                                                  if (pivot = lastID) return (NoMoreDocs)
                                                                                                                  if (pivot < curDoc)
                                                                                                           10.
                                                                                                           11.
                                                                                                                     /* pivot has already been considered, advance
                                                                                                                     one of the preceding terms */
                                                                                                           12.
                                                                                                                    aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                           13.
                                                                                                                    posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                                  else /* pivot > curDoc */
                                                                                                           14.
                                                                                                           15.
                                                                                                                    if (posting[0].DID = pivot)
                                                                                                           16.
                                                                                                                       /* Success, all terms preceding pTerm belong
                                                                                                                       to the pivot */
                                                                                                           17.
                                                                                                                       curDoc \leftarrow pivot
                                                                                                           18.
                                                                                                                       return (curDoc, posting)
                                                                                                           19.
                                                                                                                     else
                                                                                                           20.
                                                                                                                       /* not enough mass yet on pivot, advance
                                                                                                                       one of the preceding terms */
                                         D1 adalah pivot
                                                                                                                       aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                           21.
                                                                                                           22.
                                                                                                                       posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                           23.
                                                                                                                  end repeat
[hujan, 6, 1.6]
                                              D2, 0.4
                                                            D3, 0.6
                                                                          D6, 1.0
                                                                                        D8, 1.5
                                                                                                      D11, 1.6
                                                                                                                       last
                                D1, 0.7
[turun, 6, 1.5]
                                              D3, 1.0
                                                            D6, 1.5
                                                                          D8, 1.5
                                                                                        D10, 0.3
                                                                                                      D12, 1.1
                                                                                                                       last
[deras, 5, 1.8]
                                D1, 1.2
                                              D6, 1.0
                                                            D7, 0.5
                                                                         D10, 0.6
                                                                                        D11, 1.8
                                                                                                         last
```

```
1. Function next(\theta)
                                                                                                          repeat
  TOP-1
                          : [(D1,3.4)]
                                                                                                            /* Sort the terms in non decreasing order of
                                                                                                             DID */
                          : 3.4
  Threshold
                                                                                                            sort(terms, posting)
                                                                                                            /* Find pivot term - the first one with accumulated
  Pivot
                                                                                                             UB \ge \theta * /
                                                                                                            pTerm \leftarrow findPivotTerm(terms, \theta)
  curDoc
                                     #documents that are fully evaluated: 1
                                                                                                            if (pTerm = null) return (NoMoreDocs)
                                                                                                      7.
                                                                                                            pivot \leftarrow posting[pTerm].DID
                                                                                                      9.
                                                                                                            if (pivot = lastID) return (NoMoreDocs)
                                                                                                     10.
                                                                                                            if (pivot < curDoc)
                                                                                                              /* pivot has already been considered, advance
                                                                                                      11.
                                                                                                               one of the preceding terms */
                                                                                                     12.
                                                                                                              aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                     13.
                                                                                                              posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                     14.
                                                                                                            else /* pivot > curDoc */
                                                                                                     15.
                                                                                                              if (posting[0].DID = pivot)
                 D1 dikembalikan, dan dihitung score full-nya!
                                                                                                                 /* Success, all terms preceding pTerm belong
                                                                                                      16.
                                                                                                                 to the pivot */
                                                                                                                                 artinya ini dievaluasi karena
                 Lalu disimpan di daftar TOP-1 kita, dan jangan
                                                                                                                 curDoc ← pivot mengalahkan threshold
                                                                                                     17.
                 lupa update nilai threshold terbaru
                                                                                                     18.
                                                                                                                 return (curDoc, posting)
                                                                                                     19.
                                                                                                               else
                                                                                                     20.
                                                                                                                 /* not enough mass yet on pivot, advance
                                                                                                                 one of the preceding terms */
                                                                                                     21.
                                                                                                                 aterm \leftarrow pickTerm(terms[0..pTerm])
WAND(D1) = WAND(hujan, 1.6, turun, 1.5, deras, 1.8; threshold) = True
                                                                                                     22.
                                                                                                                 posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                      23.
                                                                                                            end repeat
 [hujan, 6, 1.6]
                                            D2, 0.4
                                                          D3, 0.6
                                                                                    D8, 1.5
                                                                                                 D11, 1.6
                                                                       D6, 1.0
                                                                                                                 last
 [turun, 6, 1.5]
                               D1, 0.7
                                            D3, 1.0
                                                          D6, 1.5
                                                                       D8, 1.5
                                                                                    D10, 0.3
                                                                                                 D12, 1.1
                                                                                                                 last
                               D1, 1.2
                                            D6, 1.0
                                                          D7, 0.5
                                                                      D10, 0.6
                                                                                    D11, 1.8
 [deras, 5, 1.8]
                                                                                                    last
```

```
1. Function next(\theta)
                                                                                                         2.
                                                                                                             repeat
TOP-1
                         : [(D1,3.4)]
                                                                                                               /* Sort the terms in non decreasing order of
                                                                                                                DID */
Threshold
                         : 3.4
                                                                                                               sort(terms, posting)
                                                                                                         4.
                                                                                                                /* Find pivot term - the first one with accumulated
                                                                                                         5.
Pivot
                                                                                                                UB > \theta */
                                                                                                                pTerm \leftarrow findPivotTerm(terms, \theta)
                                     #documents that are fully evaluated: 1
curDoc
                                                                                                                if (pTerm = null) return (NoMoreDocs)
                                                                                                         7.
                                                                                                                pivot \leftarrow posting[pTerm].DID
                                                                                                               if (pivot = lastID) return (NoMoreDocs)
                                                                                                         9.
                                                                                                               if (pivot < curDoc)
                                                                                                        10.
                                                                                                                  /* pivot has already been considered, advance
                                                                                                         11.
                                                                                                                  one of the preceding terms */
                                                                                                        12.
                                                                                                                  aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                                  posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                        13.
                                                                                                               else /* pivot > curDoc */
                                                                                                        14.
                                                                                                        15.
                                                                                                                  if (posting[0].DID = pivot)
                                                                                                        16.
                                                                                                                    /* Success, all terms preceding pTerm belong
                                                                                                                     to the pivot */
                                                                                                        17.
                                                                                                                    curDoc \leftarrow pivot
                                                                                                        18.
             Mulai dari awal lagi, dan lakukan sort terhadap Doc ID
                                                                                                                    return (curDoc, posting)
                                                                                                        19.
                                                                                                                  else
                                                                                                        20.
                                                                                                                    /* not enough mass yet on pivot, advance
                                                                                                                     one of the preceding terms */
                                                                                                                    aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                        21.
                                                                                                        22.
                                                                                                                    posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                        23.
                                                                                                               end repeat
                               D1, 1.5
[hujan, 6, 1.6]
                                             D2, 0.4
                                                           D3, 0.6
                                                                         D6, 1.0
                                                                                      D8, 1.5
                                                                                                    D11, 1.6
                                                                                                                     last
                               D1, 0.7
[turun, 6, 1.5]
                                             D3, 1.0
                                                           D6, 1.5
                                                                         D8, 1.5
                                                                                      D10, 0.3
                                                                                                   D12, 1.1
                                                                                                                     last
                                             D6, 1.0
                               D1, 1.2
                                                           D7, 0.5
                                                                        D10, 0.6
                                                                                      D11, 1.8
[deras, 5, 1.8]
                                                                                                       last
```

Threshold: 3.4

Pivot : 1

curDoc : 1 #documents that are fully evaluated: 1

"deras" adalah pivot term, dengan D1 adalah pivot

1.8 jadi pivot karena yang punya chance pivot itu adalah 1.6 + 1.5 + X

X Ini adalah 1.8 karena lebih dari sama dengna pivot maka dia jadi pivot term

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
             curDoc \leftarrow pivot
17.
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[hujan, 6, 1.6] [
                       D1, 1.5
                                 D2, 0.4
                                            D3, 0.6
                                                      D6, 1.0
                                                                 D8, 1.5
                                                                           D11, 1.6
                                                                                        last
[turun, 6, 1.5] |
                       D1, 0.7
                                 D3, 1.0
                                            D6, 1.5
                                                      D8, 1.5
                                                                D10, 0.3
                                                                           D12, 1.1
                                                                                        last
                       D1, 1.2
                                 D6, 1.0
                                            D7, 0.5
                                                      D10, 0.6
                                                                D11, 1.8
[deras, 5, 1.8]
                                                                             last
```

```
1. Function next(\theta)
                                                                                                       repeat
TOP-1
                        : [(D1,3.4)]
                                                                                                          /* Sort the terms in non decreasing order of
                                                                                                          DID */
                        : 3.4
Threshold
                                                                                                          sort(terms, posting)
                                                                                                          /* Find pivot term - the first one with accumulated
Pivot
                                                                                                          UB > \theta */
                                                                                                          pTerm \leftarrow findPivotTerm(terms, \theta)
curDoc
                                   #documents that are fully evaluated: 1
                                                                                                          if (pTerm = null) return (NoMoreDocs)
                                                                                                   7.
                                                                                                          pivot \leftarrow posting[pTerm].DID
                                                                                                          if (pivot = lastID) return (NoMoreDocs)
                                                                                                   9.
                                                                                                   10.
                                                                                                         if (pivot < curDoc)
                                                                                                            /* pivot has already been considered, advance
                                                                                                   11.
                                                                                                            one of the preceding terms */
                                                                                                   12.
                                                                                                            aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                   13.
                                                                                                            posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                   14.
                                                                                                          else /* pivot > curDoc */
          Advance one of the preceding terms
                                                                                                   15.
                                                                                                            if (posting[0].DID = pivot)
                                                                                                              /* Success, all terms preceding pTerm belong
                                                                                                   16.
                                                                                                               to the pivot */
                                                                                                              curDoc \leftarrow pivot
                                                                                                   17.
          pickTerm() → banyak cara untuk memilih; yang paling
                                                                                                   18.
                                                                                                              return (curDoc, posting)
          simple adalah "pilih yang pertama"
                                                                                                   19.
                                                                                                            else
                                                                                                   20.
                                                                                                              /* not enough mass yet on pivot, advance
                                                                                                               one of the preceding terms */
                                                                                                              aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                   21.
          Pointer dari term yang terpilih dimajukan hingga
                                                                                                   22.
                                                                                                              posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                   23.
                                                                                                          end repeat
          bertemu DID pertama yang >= 1 + 1 = 2
[hujan, 6, 1.6] [
                             D1, 1.5
                                          D2, 0.4
                                                        D3, 0.6
                                                                     D6, 1.0
                                                                                  D8, 1.5
                                                                                              D11, 1.6
                                                                                                               last
[turun, 6, 1.5] |
                             D1, 0.7
                                          D3, 1.0
                                                        D6, 1.5
                                                                     D8, 1.5
                                                                                 D10, 0.3
                                                                                              D12, 1.1
                                                                                                               last
                             D1, 1.2
                                          D6, 1.0
                                                        D7, 0.5
                                                                    D10, 0.6
                                                                                 D11, 1.8
[deras, 5, 1.8]
                                                                                                 last
```

Threshold : 3.4

Pivot : 1

curDoc : 1 #documents that are fully evaluated: 1

Mulai lagi dari awal. Sort berdasarkan Doc ID

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
5.
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[hujan, 6, 1.6] I
                       D1, 1.5
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
[turun, 6, 1.5]
                       D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
[deras, 5, 1.8]
                       D1, 1.2
                                 D6, 1.0
                                           D7, 0.5
                                                     D10, 0.6
                                                               D11, 1.8
                                                                            last
```

Threshold : 3.4

Pivot : 1

curDoc : 1 #documents that are fully evaluated: 1

Mulai lagi dari awal. Sort berdasarkan Doc ID

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
5.
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 0.7
                                 D3, 1.0
                                                      D8, 1.5
                                                               D10, 0.3
                                                                          D12, 1.1
[turun, 6, 1.5] |
                                           D6, 1.5
                                                                                      last
[deras, 5, 1.8]
                       D1, 1.2
                                 D6, 1.0
                                                     D10, 0.6
                                                               D11, 1.8
                                           D7, 0.5
                                                                            last
                       D1, 1.5
                                 D2, 0.4
                                                                D8, 1.5
[hujan, 6, 1.6]
                                           D3, 0.6
                                                      D6, 1.0
                                                                          D11, 1.6
                                                                                      last
```

Threshold: 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Mulai lagi dari awal. Sort berdasarkan Doc ID

$$1.5 + 1.8 + 1.6 >= 3.4$$

Pivot term = hujan, pivot = D2

```
1. Function next(\theta)
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
                    - lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
              curDoc \leftarrow pivot
17.
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5] D1, 0.7 D3, 1.0 D6, 1.5 D8, 1.5 D10, 0.3 D12, 1.1 last
```

```
[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last
```

```
[hujan, 6, 1.6] D1, 1.5 D2, 0.4 D3, 0.6 D6, 1.0 D8, 1.5 D11, 1.6 last
```

Threshold : 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Not enough mass yet on pivot, advance one of the preceding terms.

pickTerm() → kita pilih yang pertama

```
1. Function next(\theta)
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \ge \theta * /
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        ena repeat
```

```
D1, 0.7
                                 D3, 1.0
                                                      D8, 1.5
                                                                D10, 0.3
                                                                          D12, 1.1
[turun, 6, 1.5] |
                                            D6, 1.5
                                                                                       last
[deras, 5, 1.8]
                       D1, 1.2
                                                     D10, 0.6
                                                                D11, 1.8
                                 D6, 1.0
                                            D7, 0.5
                                                                             last
                       D1, 1.5
[hujan, 6, 1.6] [
                                 D2, 0.4
                                            D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
                                                                                       last
```

Threshold : 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Not enough mass yet on pivot, advance one of the preceding terms.

pickTerm() → kita pilih yang pertama Lalu kita pindahkan ke posisi pertama yang Doc ID >= pivot

```
1. Function next(\theta)
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        ena repeat
```

```
D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                      D8, 1.5
                                                                D10, 0.3
                                                                          D12, 1.1
[turun, 6, 1.5]
                                                                                       last
                                 D6, 1.0
                                                     D10, 0.6
                                                                D11, 1.8
[deras, 5, 1.8]
                       D1, 1.2
                                            D7, 0.5
                                                                            last
                       D1, 1.5
                                 D2, 0.4
                                           D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
[hujan, 6, 1.6] [
                                                                                       last
```

Threshold : 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Mulai lagi dari awal, lakukan sorting

```
1. Function next(\theta)
2.
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
5.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                      D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
[deras, 5, 1.8]
                       D1, 1.2
                                                    D10, 0.6
                                                               D11, 1.8
                                 D6, 1.0
                                           D7, 0.5
                                                                           last
                       D1, 1.5
                                 D2, 0.4
                                                               D8, 1.5
[hujan, 6, 1.6]
                                           D3, 0.6
                                                     D6, 1.0
                                                                         D11, 1.6
                                                                                      last
```

Threshold : 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Mulai lagi dari awal, lakukan sorting

```
1. Function next(\theta)
2.
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
5.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.2
                                                    D10, 0.6
                                                              D11, 1.8
[deras, 5, 1.8]
                                D6, 1.0
                                           D7, 0.5
                                                                           last
                      D1, 1.5
                                D2, 0.4
[hujan, 6, 1.6] |
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                     last
                      D1, 0.7
                                           D6, 1.5
[turun, 6, 1.5]
                                D3, 1.0
                                                     D8, 1.5
                                                              D10, 0.3
                                                                         D12, 1.1
                                                                                     last
```

1. Function  $next(\theta)$ repeat TOP-1 : [(D1,3.4)] DID \*/ Threshold : 3.4 Pivot : 2  $UB > \theta */$ 6. curDoc #documents that are fully evaluated: 1 7. 9. 10. 11. 12. 13. 14. Pivot term = hujan dengan pivot = D2 15. 16. 17.

```
/* Sort the terms in non decreasing order of
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
        pivot \leftarrow posting[pTerm].DID
                    - lastID) return (NoMoreDocs)
        if (pivot < curDoc)
           /* pivot has already been considered, advance
           one of the preceding terms */
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
              to the pivot */
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.2
                                                               D11, 1.8
[deras, 5, 1.8]
                                 D6, 1.0
                                           D7, 0.5
                                                    D10, 0.6
                                                                           last
[hujan, 6, 1.6] [
                      D1, 1.5
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                     last
[turun, 6, 1.5]
                      D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                     last
```

TOP-1 : [(D1,3.4)] Threshold : 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Kita majukan pointer di deras ke Doc ID pertama yang >= pivot.

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           معام
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.2
                                                               D11, 1.8
[deras, 5, 1.8]
                                 D6, 1.0
                                           D7, 0.5
                                                     D10, 0.6
                                                                            last
                       D1, 1.5
[hujan, 6, 1.6] [
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                                           D6, 1.5
[turun, 6, 1.5]
                      D1, 0.7
                                 D3, 1.0
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
```

: 3.4 Threshold

Pivot : 2

curDoc #documents that are fully evaluated: 1

> Kita majukan pointer di deras ke Doc ID pertama yang >= pivot.

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           معام
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.2
                                                               D11, 1.8
[deras, 5, 1.8]
                                 D6, 1.0
                                           D7, 0.5
                                                     D10, 0.6
                                                                            last
                       D1, 1.5
[hujan, 6, 1.6] [
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                                           D6, 1.5
[turun, 6, 1.5]
                      D1, 0.7
                                 D3, 1.0
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
```

Threshold : 3.4

Pivot : 2

curDoc : 1 #documents that are fully evaluated: 1

Sort berdasarkan Doc ID

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[hujan, 6, 1.6] I
                       D1, 1.5
                                 D2, 0.4
                                           D3, 0.6
                                                                D8, 1.5
                                                                          D11, 1.6
                                                      D6, 1.0
                                                                                      last
[turun, 6, 1.5]
                       D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                     D8, 1.5
                                                               D10, 0.3
                                                                          D12, 1.1
                                                                                      last
                                 D6, 1.0
                                           D7, 0.5
[deras, 5, 1.8]
                       D1, 1.2
                                                     D10, 0.6
                                                               D11, 1.8
                                                                            last
```

Threshold : 3.4

Pivot : 6

curDoc #documents that are fully evaluated: 1

Pivot term = deras, dengan pivot = D6

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[hujan, 6, 1.6] I
                       D1, 1.5
                                 D2, 0.4
                                           D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
                                                                                       last
[turun, 6, 1.5]
                       D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                      D8, 1.5
                                                                D10, 0.3
                                                                          D12, 1.1
                                                                                       last
                                           D7, 0.5
[deras, 5, 1.8]
                       D1, 1.2
                                 D6, 1.0
                                                     D10, 0.6
                                                                D11, 1.8
                                                                            last
```

Threshold: 3.4

Pivot : 6

[deras, 5, 1.8]

curDoc : 1 #documents that are fully evaluated: 1

Not enough mass yet on pivot, advance one of the preceding terms.

pickTerm() → kita pilih yang pertama Lalu kita pindahkan ke posisi pertama yang Doc ID >= pivot

D6, 1.0

D1, 1.2

```
[hujan, 6, 1.6] |
                       D1, 1.5
                                  D2, 0.4
                                             D3, 0.6
                                                       D6, 1.0
                                                                  D8, 1.5
                                                                            D11, 1.6
                                                                                         last
[turun, 6, 1.5] |
                       D1, 0.7
                                  D3, 1.0
                                             D6, 1.5
                                                       D8, 1.5
                                                                 D10, 0.3
                                                                            D12, 1.1
                                                                                         last
```

D7, 0.5

D10, 0.6

D11, 1.8

```
repeat
          /* Sort the terms in non decreasing order of
           DID */
          sort(terms, posting)
          /* Find pivot term - the first one with accumulated
           UB \ge \theta * /
          pTerm \leftarrow findPivotTerm(terms, \theta)
          if (pTerm = null) return (NoMoreDocs)
  7.
          pivot \leftarrow posting[pTerm].DID
          if (pivot = lastID) return (NoMoreDocs)
  9.
  10.
          if (pivot < curDoc)
             /* pivot has already been considered, advance
  11.
             one of the preceding terms */
  12.
             aterm \leftarrow pickTerm(terms[0..pTerm])
  13.
             posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
  14.
          else /* pivot > curDoc */
  15.
             if (posting[0].DID = pivot)
                /* Success, all terms preceding pTerm belong
  16.
                to the pivot */
               curDoc \leftarrow pivot
  17.
  18.
                return (curDoc, posting)
  19.
             else
  20.
                /* not enough mass yet on pivot, advance
                one of the preceding terms */
                aterm \leftarrow pickTerm(terms[0..pTerm])
  21.
  22.
                posting[aterm] \leftarrow aterm.iterator.next(pivot)
  23.
          end repeat
last
```

1. Function  $next(\theta)$ 

Threshold : 3.4

Pivot : 6

curDoc : 1 #documents that are fully evaluated: 1

D3 di-skip karena dijamin tidak akan memberikan score yang > 3.4. Mengapa? ☺

Catat bahwa posting list terakhir berada pada posisi D6. Seandainya dua posting list pertama berada pada posisi D3, paling tinggi hanya 1.6 + 1.5 = 3.1 yang < 3.4.

```
1. Function next(\theta)
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \ge \theta * /
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
14.
        else /* pivot > curDoc */
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
21.
              aterm \leftarrow pickTerm(terms[0..pTerm])
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[hujan, 6, 1.6] [
                       D1, 1.5
                                 D2, 0.4
                                            D3, 0.6
                                                                 D8, 1.5
                                                                           D11, 1.6
                                                      D6, 1.0
                                                                                       last
[turun, 6, 1.5] |
                       D1, 0.7
                                 D3, 1.0
                                            D6, 1.5
                                                      D8, 1.5
                                                                D10, 0.3
                                                                           D12, 1.1
                                                                                       last
                       D1, 1.2
[deras, 5, 1.8]
                                 D6, 1.0
                                            D7, 0.5
                                                      D10, 0.6
                                                                D11, 1.8
                                                                             last
```

Threshold : 3.4

Pivot : 6

curDoc : 1 #documents that are fully evaluated: 1

Sort dokumen berdasarkan Doc ID

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
5.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 0.7
                                 D3, 1.0
                                                     D8, 1.5
[turun, 6, 1.5]
                                           D6, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                     last
                      D1, 1.5
                                 D2, 0.4
[hujan, 6, 1.6] |
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                     last
                                           D7, 0.5
[deras, 5, 1.8]
                      D1, 1.2
                                 D6, 1.0
                                                    D10, 0.6
                                                               D11, 1.8
                                                                           last
```

Threshold : 3.4

Pivot : 6

curDoc : 1 #documents that are fully evaluated: 1

Pivot term = deras, dengan pivot = 6

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                      D1, 0.7
                                 D3, 1.0
                                                     D8, 1.5
                                           D6, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
                       D1, 1.5
                                 D2, 0.4
[hujan, 6, 1.6] |
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                                           D7, 0.5
[deras, 5, 1.8]
                       D1, 1.2
                                 D6, 1.0
                                                     D10, 0.6
                                                               D11, 1.8
                                                                            last
```

Threshold : 3.4

Pivot : 6

curDoc : 1 #documents that are fully evaluated: 1

Majukan Doc ID di term turun ke Doc ID pertama yang >= pivot

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                       D1, 0.7
                                 D3, 1.0
                                                      D8, 1.5
                                           D6, 1.5
                                                                D10, 0.3
                                                                          D12, 1.1
                                                                                       last
                       D1, 1.5
                                 D2, 0.4
[hujan, 6, 1.6] |
                                           D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
                                                                                       last
```

```
[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last
```

Threshold: 3.4

Pivot : 6

curDoc : 1 #documents that are fully evaluated: 1

Majukan Doc ID di term turun ke Doc ID pertama yang >= pivot

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                                                      D8, 1.5
                       D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
                       D1, 1.5
                                 D2, 0.4
[hujan, 6, 1.6] |
                                           D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                       D1, 1.2
                                           D7, 0.5
[deras, 5, 1.8]
                                 D6, 1.0
                                                     D10, 0.6
                                                               D11, 1.8
                                                                            last
```

Threshold : 3.4

Pivot : 6

[deras, 5, 1.8]

D1, 1.2

D6, 1.0

curDoc : 1

#documents that are fully evaluated: 1

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
        DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                       D1, 0.7
                                 D3, 1.0
                                                      D8, 1.5
                                                                D10, 0.3
                                                                          D12, 1.1
                                            D6, 1.5
                                                                                       last
                       D1, 1.5
[hujan, 6, 1.6] |
                                 D2, 0.4
                                            D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
                                                                                       last
```

D7, 0.5

D10, 0.6

D11, 1.8

last

Threshold : 3.4

Pivot : 6

curDoc : 1 #documents that are fully evaluated: 1

Artinya pivot ID muncul di semua preceding lists!

Dokumen pivot dievaluasi full = 1.5 + 1.0 + 1.0 = 3.5

3.5 > 3.4!

```
1. Function next(\theta)
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \ge \theta * /
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
           if (posting[0].DID = pivot)
15.
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
             curDoc \leftarrow pivot
17.
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                      D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
                       D1, 1.5
                                 D2, 0.4
[hujan, 6, 1.6] |
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                      D1, 1.2
                                           D7, 0.5
[deras, 5, 1.8]
                                 D6, 1.0
                                                     D10, 0.6
                                                               D11, 1.8
                                                                            last
```

```
1. Function next(\theta)
                                                                                                          repeat
  TOP-1
                          : [(D6,3.5)]
                                                                                                             /* Sort the terms in non decreasing order of
                                                                                                             DID */
  Threshold
                          : 3.5
                                                                                                             sort(terms, posting)
                                                                                                             /* Find pivot term - the first one with accumulated
  Pivot
                          : 6
                                                                                                             UB \ge \theta * /
                                                                                                             pTerm \leftarrow findPivotTerm(terms, \theta)
  curDoc
                                     #documents that are fully evaluated: 2
                          : 6
                                                                                                      7.
                                                                                                             if (pTerm = null) return (NoMoreDocs)
                                                                                                             pivot \leftarrow posting[pTerm].DID
                                                                                                             if (pivot = lastID) return (NoMoreDocs)
                                                                                                      9.
                                                                                                      10.
                                                                                                             if (pivot < curDoc)
                                                                                                               /* pivot has already been considered, advance
                                                                                                      11.
                                                                                                                one of the preceding terms */
              Artinya pivot ID muncul di semua preceding lists!
                                                                                                      12.
                                                                                                               aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                      13.
                                                                                                               posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                      14.
                                                                                                             else /* pivot > curDoc */
              Dokumen pivot dievaluasi full = 1.5 + 1.0 + 1.0 = 3.5
                                                                                                      15.
                                                                                                               if (posting[0].DID = pivot)
                                                                                                                  /* Success, all terms preceding pTerm belong
                                                                                                      16.
                                                                                                                  to the pivot */
                                                                                                                 curDoc \leftarrow pivot
                                                                                                      17.
              3.5 > 3.4
                                                                                                      18.
                                                                                                                 return (curDoc, posting)
                                                                                                      19.
                                                                                                               else
                                                                                                      20.
                                                                                                                 /* not enough mass yet on pivot, advance
                                                                                                                  one of the preceding terms */
                                                                                                      21.
                                                                                                                 aterm \leftarrow pickTerm(terms[0..pTerm])
WAND(D6) = WAND(hujan, 1.6, turun, 1.5, deras, 1.8; threshold) = True
                                                                                                      22.
                                                                                                                 posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                             end repeat
  [turun, 6, 1.5]
                               D1, 0.7
                                             D3, 1.0
                                                          D6, 1.5
                                                                       D8, 1.5
                                                                                    D10, 0.3
                                                                                                 D12, 1.1
                                                                                                                  last
                               D1, 1.5
                                             D2, 0.4
  [hujan, 6, 1.6] |
                                                          D3, 0.6
                                                                       D6, 1.0
                                                                                    D8, 1.5
                                                                                                 D11, 1.6
                                                                                                                  last
                               D1, 1.2
  [deras, 5, 1.8]
                                             D6, 1.0
                                                          D7, 0.5
                                                                       D10, 0.6
                                                                                    D11, 1.8
                                                                                                    last
```

TOP-1 : [(D6,3.5)]

Threshold : 3.5

Pivot : 6

curDoc : 6

#documents that are fully evaluated: 2

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB \ge \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
7.
        if (pTerm = null) return (NoMoreDocs)
        pivot \leftarrow posting[pTerm].DID
8.
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5]
                                 D3, 1.0
                                                     D8, 1.5
                       D1, 0.7
                                           D6, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
                       D1, 1.5
[hujan, 6, 1.6] |
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                                D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                                           D7, 0.5
[deras, 5, 1.8]
                       D1, 1.2
                                 D6, 1.0
                                                     D10, 0.6
                                                               D11, 1.8
                                                                            last
```

1. Function  $next(\theta)$ repeat TOP-1 : [(D6,3.5)] /\* Sort the terms in non decreasing order of DID \*/ Threshold : 3.5 sort(terms, posting) /\* Find pivot term - the first one with accumulated Pivot : 6  $UB \ge \theta * /$ pTerm  $\leftarrow$  findPivotTerm(terms,  $\theta$ ) curDoc #documents that are fully evaluated: 2 : 6 if (pTerm = null) return (NoMoreDocs) 7.  $pivot \leftarrow posting[pTerm].DID$ 9. **if** (pivot = lastID) **return** (NoMoreDocs) 10. if (pivot < curDoc) /\* pivot has already been considered, advance 11. one of the preceding terms \*/ 12.  $aterm \leftarrow pickTerm(terms[0..pTerm])$ 13.  $posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)$ 14. else /\* pivot > curDoc \*/ 15. **if** (posting[0].DID = pivot)/\* Success, all terms preceding pTerm belong 16. Pointer dari term yang terpilih dimajukan hingga to the pivot \*/ 17.  $curDoc \leftarrow pivot$ bertemu DID pertama yang >= 6 + 1 = 7 18. return (curDoc, posting) 19. else 20. /\* not enough mass yet on pivot, advance one of the preceding terms \*/ Yaitu D8  $aterm \leftarrow pickTerm(terms[0..pTerm])$ 21. 22.  $posting[aterm] \leftarrow aterm.iterator.next(pivot)$ 23. end repeat D1, 0.7 [turun, 6, 1.5] D3, 1.0 D6, 1.5 D8, 1.5 D10, 0.3 D12, 1.1 last D1, 1.5 D2, 0.4 [hujan, 6, 1.6] D3, 0.6 D6, 1.0 D8, 1.5 D11, 1.6 last

[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last

TOP-1 : [(D6,3.5)] Threshold : 3.5

Pivot : 8

curDoc : 6

#documents that are fully evaluated: 2

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.5
[hujan, 6, 1.6] [
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
[deras, 5, 1.8]
                       D1, 1.2
                                                     D10, 0.6
                                                               D11, 1.8
                                 D6, 1.0
                                           D7, 0.5
                                                                            last
                       D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
[turun, 6, 1.5]
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
```

TOP-1 : [(D6,3.5)] Threshold : 3.5

Pivot : 8

curDoc : 6 #documents t

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \ge \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
21.
              aterm \leftarrow pickTerm(terms[0..pTerm])
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.5
                                 D2, 0.4
[hujan, 6, 1.6] [
                                           D3, 0.6
                                                     D6, 1.0
                                                                D8, 1.5
                                                                         D11, 1.6
                                                                                      last
[deras, 5, 1.8]
                       D1, 1.2
                                                     D10, 0.6
                                                               D11, 1.8
                                 D6, 1.0
                                           D7, 0.5
                                                                            last
                       D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
[turun, 6, 1.5]
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                      last
```

Threshold : 3.5

Pivot : 8

curDoc : 6

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[deras, 5, 1.8]
                      D1, 1.2
                                 D6, 1.0
                                           D7, 0.5
                                                    D10, 0.6
                                                               D11, 1.8
                                                                           last
                                 D2, 0.4
[hujan, 6, 1.6] I
                      D1, 1.5
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                     last
                      D1, 0.7
                                 D3, 1.0
                                           D6, 1.5
[turun, 6, 1.5]
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                                                                     last
```

```
2.
                                                                                                              repeat
TOP-1
                          : [(D6,3.5)]
                                                                                                                 /* Sort the terms in non decreasing order of
                                                                                                                 DID */
Threshold
                          : 3.5
                                                                                                                 sort(terms, posting)
                                                                                                                 /* Find pivot term - the first one with accumulated
Pivot
                          : 8
                                                                                                                 UB \ge \theta */
                                                                                                                 pTerm \leftarrow findPivotTerm(terms, \theta)
curDoc
                                     #documents that are fully evaluated: 2
                          : 6
                                                                                                         7.
                                                                                                                 if (pTerm = null) return (NoMoreDocs)
                                                                                                                 pivot \leftarrow posting[pTerm].DID
                                                                                                                 if (pivot = lastID) return (NoMoreDocs)
                                                                                                          9.
                                                                                                          10.
                                                                                                                 if (pivot < curDoc)
                                                                                                                   /* pivot has already been considered, advance
                                                                                                          11.
                                                                                                                    one of the preceding terms */
                                                                                                          12.
                                                                                                                   aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                                   posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
                                                                                                          13.
                                                                                                                 else /* pivot > curDoc */
                                                                                                          14.
                                                                                                          15.
                                                                                                                   if (posting[0].DID = pivot)
                                                                                                          16.
                                                                                                                      /* Success, all terms preceding pTerm belong
                                                                                                                      to the pivot */
                                                                                                          17.
                                                                                                                     curDoc \leftarrow pivot
                                                                                                          18.
                                                                                                                     return (curDoc, posting)
                                                                                                          19.
                                                                                                                   else
                                                                                                          20.
                                                                                                                      /* not enough mass yet on pivot, advance
                                                                                                                      one of the preceding terms */
                                                                                                                      aterm \leftarrow pickTerm(terms[0..pTerm])
                                                                                                          21.
                                                                                                          22.
                                                                                                                      posting[aterm] \leftarrow aterm.iterator.next(pivot)
                                                                                                          23.
                                                                                                                 end repeat
[deras, 5, 1.8]
                               D1, 1.2
                                             D6, 1.0
                                                           D7, 0.5
                                                                        D10, 0.6
                                                                                       D11, 1.8
                                                                                                        last
                                             D2, 0.4
[hujan, 6, 1.6] I
                               D1, 1.5
                                                           D3, 0.6
                                                                         D6, 1.0
                                                                                       D8, 1.5
                                                                                                     D11, 1.6
                                                                                                                      last
                               D1, 0.7
                                             D3, 1.0
[turun, 6, 1.5]
                                                           D6, 1.5
                                                                         D8, 1.5
                                                                                       D10, 0.3
                                                                                                    D12, 1.1
                                                                                                                      last
```

1. Function  $next(\theta)$ 

Threshold : 3.5

Pivot : 10

curDoc : 6 #

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
        DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
16.
              /* Success, all terms preceding pTerm belong
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
21.
              aterm \leftarrow pickTerm(terms[0..pTerm])
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

[hujan, 6, 1.6]	D1, 1.5	D2, 0.4	D3, 0.6	D6, 1.0	D8, 1.5	D11, 1.6	last
. [							
[turun, 6, 1.5]	D1, 0.7	D3, 1.0	D6, 1.5	D8, 1.5	D10, 0.3	D12, 1.1	last
[deras, 5, 1.8]	D1, 1.2	D6, 1.0	D7, 0.5	D10, 0.6	D11, 1.8	last	

TOP-1 : [(D6,3.5)] Threshold : 3.5

**Pivot** : 10

curDoc : 6 #documents that are fully evaluated: 2

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \ge \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           _{
m else}
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.5
[hujan, 6, 1.6] I
                                 D2, 0.4
                                                                D8, 1.5
                                                                          D11, 1.6
                                            D3, 0.6
                                                      D6, 1.0
                                                                                       last
[turun, 6, 1.5] |
                       D1, 0.7
                                 D3, 1.0
                                            D6, 1.5
                                                      D8, 1.5
                                                                D10, 0.3
                                                                          D12, 1.1
                                                                                       last
[deras, 5, 1.8]
                       D1, 1.2
                                 D6, 1.0
                                            D7, 0.5
                                                     D10, 0.6
                                                                D11, 1.8
                                                                             last
```

Threshold : 3.5

**Pivot** : 11

[hujan, 6, 1.6]

curDoc : 6 #documents that are fully evaluated: 2

D1, 1.5

D2, 0.4

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[turun, 6, 1.5] D1, 0.7 D3, 1.0 D6, 1.5 D8, 1.5 D10, 0.3 D12, 1.1 last [deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last
```

D3, 0.6

D8, 1.5

D11, 1.6

last

D6, 1.0

TOP-1 : [(D6,3.5)] Threshold : 3.5

Pivot : 11

curDoc : 6 #documents that are fully evaluated: 2

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \ge \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
11.
           /* pivot has already been considered, advance
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           _{
m else}
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 0.7
                                 D3, 1.0
                                                               D10, 0.3
                                                                          D12, 1.1
[turun, 6, 1.5] |
                                           D6, 1.5
                                                      D8, 1.5
                                                                                      last
                                                     D10, 0.6
[deras, 5, 1.8]
                                 D6, 1.0
                       D1, 1.2
                                           D7, 0.5
                                                               D11, 1.8
                                                                            last
                                                                D8, 1.5
[hujan, 6, 1.6]
                       D1, 1.5
                                 D2, 0.4
                                           D3, 0.6
                                                      D6, 1.0
                                                                          D11, 1.6
                                                                                      last
```

Threshold : 3.5

Pivot : 12

curDoc : 6 #doc

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
11.
           /* pivot has already been considered, advance
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last
```

```
[hujan, 6, 1.6] D1, 1.5 D2, 0.4 D3, 0.6 D6, 1.0 D8, 1.5 D11, 1.6 last
```

```
[turun, 6, 1.5] D1, 0.7 D3, 1.0 D6, 1.5 D8, 1.5 D10, 0.3 D12, 1.1 last
```

Threshold : 3.5

Pivot : 12

curDoc : 6

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
13.
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.2
                                 D6, 1.0
                                                    D10, 0.6
                                                               D11, 1.8
[deras, 5, 1.8]
                                           D7, 0.5
                                                                           last
                      D1, 1.5
[hujan, 6, 1.6]
                                 D2, 0.4
                                           D3, 0.6
                                                     D6, 1.0
                                                               D8, 1.5
                                                                         D11, 1.6
                                                                                      last
                       D1, 0.7
                                 D3, 1.0
                                                     D8, 1.5
                                                               D10, 0.3
                                                                         D12, 1.1
                                           D6, 1.5
[turun, 6, 1.5]
                                                                                      last
```

Threshold : 3.5

Pivot : last

curDoc : 6 #documents that are fully evaluated: 2

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
4.
        /* Find pivot term - the first one with accumulated
         UB > \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
6.
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
        if (pivot < curDoc)
10.
11.
           /* pivot has already been considered, advance
            one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
              /* Success, all terms preceding pTerm belong
16.
              to the pivot */
17.
              curDoc \leftarrow pivot
18.
              return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
              aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
```

23.

end repeat

```
D1, 1.5
[hujan, 6, 1.6] [
                                 D2, 0.4
                                            D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
                                                                                       last
                                 D3, 1.0
[turun, 6, 1.5]
                       D1, 0.7
                                                      D8, 1.5
                                                                D10, 0,3
                                                                          D12, 1.1
                                            D6, 1.5
                                                                                       last
```

[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last

Threshold : 3.5

Pivot : last

curDoc : 6 #documents that are fully evaluated: 2

STOP!

```
1. Function next(\theta)
2.
     repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB \geq \theta */
        pTerm \leftarrow findPivotTerm(terms, \theta)
        if (pTerm = null) return (NoMoreDocs)
7.
        pivot \leftarrow posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
9.
10.
        if (pivot < curDoc)
           /* pivot has already been considered, advance
11.
           one of the preceding terms */
12.
           aterm \leftarrow pickTerm(terms[0..pTerm])
           posting[aterm] \leftarrow aterm.iterator.next(curDoc+1)
13.
        else /* pivot > curDoc */
14.
15.
           if (posting[0].DID = pivot)
16.
              /* Success, all terms preceding pTerm belong
              to the pivot */
17.
             curDoc \leftarrow pivot
18.
             return (curDoc, posting)
19.
           else
20.
              /* not enough mass yet on pivot, advance
              one of the preceding terms */
             aterm \leftarrow pickTerm(terms[0..pTerm])
21.
22.
              posting[aterm] \leftarrow aterm.iterator.next(pivot)
23.
        end repeat
```

```
D1, 1.5
[hujan, 6, 1.6] [
                                 D2, 0.4
                                            D3, 0.6
                                                      D6, 1.0
                                                                D8, 1.5
                                                                          D11, 1.6
                                                                                       last
                                 D3, 1.0
[turun, 6, 1.5]
                       D1, 0.7
                                                      D8, 1.5
                                                                D10, 0,3
                                                                          D12, 1.1
                                            D6, 1.5
                                                                                       last
```

[deras, 5, 1.8] D1, 1.2 D6, 1.0 D7, 0.5 D10, 0.6 D11, 1.8 last

TOP-2:
Threshold:
Pivot:
curDoc:

## Latihan

Lakukan hal yang sama dengan sebelumnya. Namun sekarang Top-2 yang diambil!

