Temu Kembali Informasi/ Information Retrieval

Table Of Content

Boolean Retrieval Model

- Model proses pencarian informasi dari query, yang menggunakan ekspresi boolean.
- Ekspresi boolean dapat berupa operator logika AND, OR dan NOT.
- Hasil perhitungannya hanya berupa nilai **binary** (1 atau 0).
- Ini menyebabkan di dalam Boolean Retrieval Model (BRM), yang ada hanya dokumen relevan atau tidak sama sekali. Tidak ada pertimbangan dokumen yang ‘mirip’.

Boolean Retrieval Model

- Dalam pengerjaan operator boolean (AND, NOT, OR) ada urutan pengerjaannya (Operator precedence).
- Urutannya adalah:
  - () → Prioritas yang berada dalam tanda kurung
  - NOT
  - AND
  - OR
- Jadi kalau ada query sebagai berikut?
  - (Madding OR crow) AND Killed OR slain
  - (Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

Permasalahan IR

- Misalkan kita ingin mencari dari cerita-cerita karangan shakespeare yang mengandung kata **Brutus AND Caesar AND NOT Calpurnia**.
- Salah satu cara adalah: Baca semua teks yang ada dari awal sampai akhir.
- Komputer juga bisa disuruh melakukan hal ini (menggantikan manusia). Proses ini disebut ***grepping.***
- Melihat kemajuan komputer jaman sekarang, *grepping* bisa jadi solusi yang baik.

Permasalahan IR

- Tapi, kalau sudah bicara soal ribuan dokumen, kita perlu melakukan sesuatu yang lebih baik.
- Karena ada beberapa tuntutan yang harus dipenuhi :
  - Kecepatan dalam pemrosesan dokumen yang jumlahnya sangat banyak.
  - Fleksibilitas.
  - Perangkingan.
- Salah satu cara pemecahannya adalah dengan membangun **index** dari dokumen.

Incidence Matrix

- Incidence matrix adalah suatu matrix yang terdiri dari kolom (dokumen) dan baris (token/terms/kata).
- Pembangunan index akan berbeda untuk tiap metode Retrieval.
- Untuk boolean model, salah satunya kita akan menggunakan Incidence matrix sebagai index dari korpus (kumpulan dokumen) data kita.
- Dokumen yang ada di kolom adalah semua dokumen yang terdapat pada korpus data kita.

Incidence Matrix

- **Token/Terms/Kata pada baris** adalah semua token unik (kata yang berbeda satu dengan yang lainnya) dalam seluruh dokumen yang ada.
- Saat suatu token(*t*) ada dalam dokumen(*d*), maka nilai dari baris dan kolom (*t,d*) adalah **1**. Jika tidak ditemukan, maka nilai kolom (*t,d*) adalah **0**.
- Dari sudut pandang kolom, kita bisa tahu token apa saja yang ada di satu dokumen (*d*).
- Dari sudut pandang barisnya, kita bisa tahu di dokumen mana saja token (*t*) ada (*posting lists*).

Case Study A (1 of 3)

- Perhatikan tabel berikut. (Vektor baris menyatakan keberadaan suatu Token/Terms/Kata unik yang ada dalam semua dokumen. Vektor kolom menyatakan semua nama dokumen yang digunakan). Diketahui 6 dokumen dengan masing-masing kata yang terdapat di dalamnya. Jika kata tersebut berada dalam dokumen, maka Term Frekuensi Biner/ $TF_{biner} = 1$, jika tidak $TF_{biner} = 0$.

	Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
Mercy	1	0	1	1	1	1
Worser	1	0	1	1	1	0
….						

Case Study A (2 of 3)

- Dengan mengunakan Incidence matrix yang sudah dibangun, kita sudah bisa memecahkan masalah yang pertama dihadapi tadi.
- Kemudian misalkan mencari hasil Boolean Query Retrieval : **Brutus AND Caesar AND NOT Calpurnia**
- Maka dapat diketahui dengan mudah, dokumen mana saja yang mengandung kata Brutus dan Caesar, tetapi tidak mengandung kata Calpurnia.

Case Study A (3 of 3)

- TF_{biner}(Brutus) = 110100
- TF_{biner}(Caesar) = 110111
- TF_{biner}(Calpurnia) = 010000
- Brutus AND Caesar AND NOT Calpurnia
  = 110100 AND 110111 AND NOT 010000
  = 110100 AND 110111 AND 101111
  = **100100**

	Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
Mercy	1	0	1	1	1	1
Worser	1	0	1	1	1	0
....						

1	0	0	1	0	0
Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth

- Berarti, jawaban hasil Boolean Query Retrieval : **Brutus AND Caesar AND NOT Calpurnia** adalah Dokumen *"Antony & Cleopatra"* dan *"Hamlet"*

Latihan Individu (Today)

- Buatlah Incidence matrix untuk dokumen-dokumen berikut :

Dokumen (Doc)	Isi (Content)
Doc 1	New home sales top forecasts
Doc 2	Home sales rise in july
Doc 3	Increase in home sales in july
Doc 4	July new home sales rise

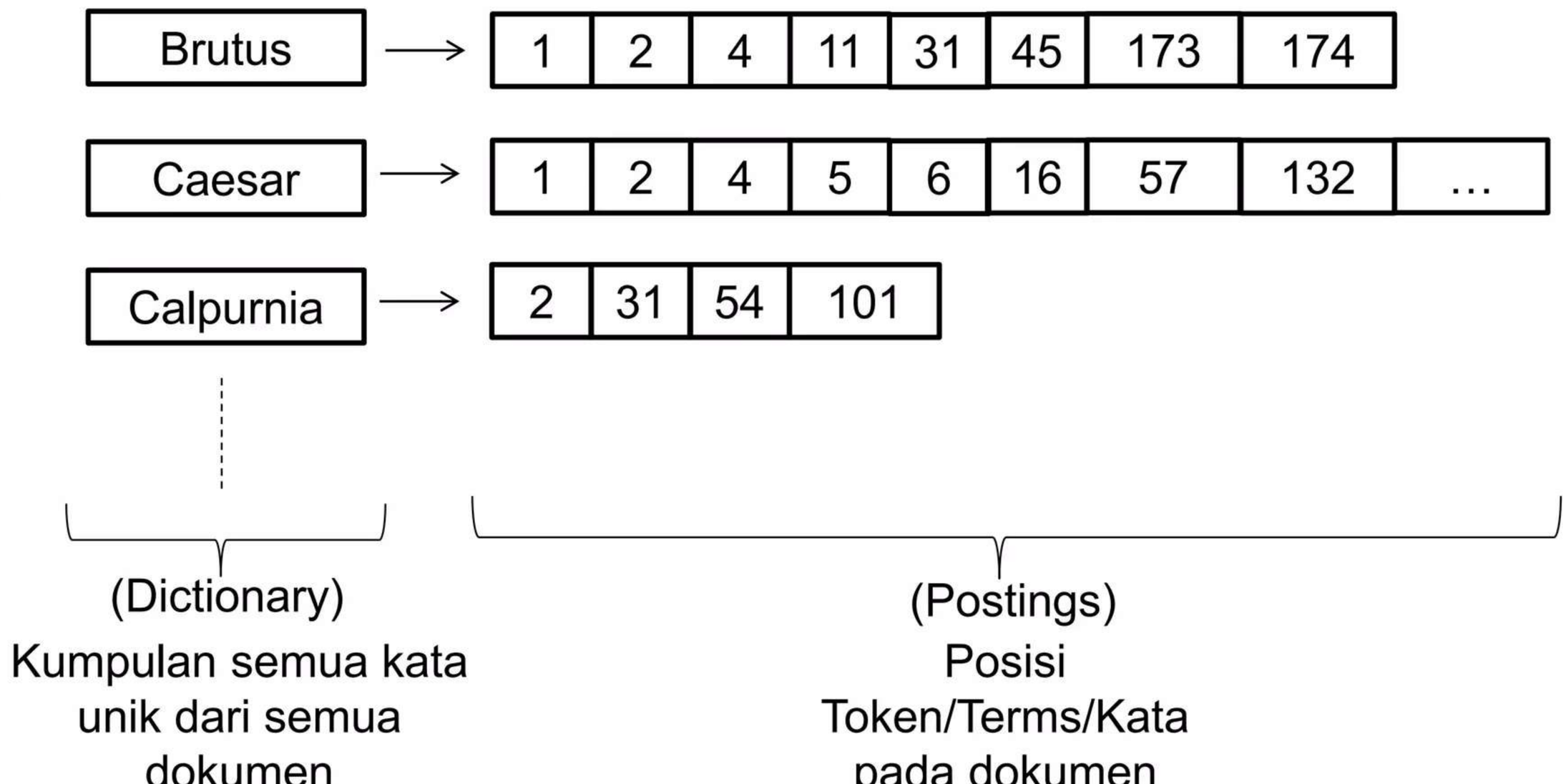
- Tentukan hasil boolean query retrieval berikut berdasarkan Incidence matrix di atas :
  - Home AND Sales AND NOT July
  - Home AND July AND NOT Sales

Inverted Index

- Inverted index adalah sebuah struktur data index yang dibangun untuk memudahkan query pencarian yang memotong tiap kata (term) yang berbeda dari suatu daftar term dokumen.
- Tujuan :
  - Meningkatkan kecepatan dan efisiensi dalam melakukan pencarian pada sekumpulan dokumen.
  - Menemukan dokumen-dokumen yang mengandung query user.

Inverted Index

- Ilustrasi :



Inverted Index

- Inverted index mempunyai vocabulary, yang berisi seluruh term yang berbeda pada masing-masing dokumennya (unik), dan tiap-tiap term yang berbeda ditempatkan pada *inverted list*.
- Notasi :

<id_j, f_{ij},[O_1,O_2,…,O_k | f_{ij} |]>

Keterangan :

– id_j adalah ID dokumen d_j yang mengandung term t_i

– f_{ij} adalah frekuensi kemunculan term t_i didokumen d_j

– O_k adalah posisi term t_i di dokumen d_j.

Case Study B (1 of 4)

- Perhatikan beberapa dokumen berikut : (Buatlah Inverted Index-nya)
  - Dokument 1 (Id1):

Algoritma	Genetik	dapat	digunakan	untuk
1	2	3	4	5

Optimasi	fuzzy
6	7

  - Dokument 2 (Id2) :

Optimasi	fungsi	keanggotaan	pada	fuzzy
1	2	3	4	5

  - Dokument 3 (Id3) :

Algoritma	genetik	merupakan	algoritma	Learning
1	2	3	4	5

Case Study B (2 of 4)

- Set vocabulary :
  {algoritma, genetik, dapat, digunakan, untuk, optimasi, fuzzy, fungsi, keanggotaan, pada, merupakan, learning}
- Inverted Index sederhana :

Term	Inverted List
Algoritma	Id1, id3
Dapat	Id1
Digunakan	Id1
Fungsi	Id2
Fuzzy	Id1, id2
Genetik	Id1, id3
Keanggotaan	Id2
Learning	Id3
Merupakan	Id3
Optimasi	Id1, id2
Pada	Id2
Untuk	id1

Case Study B (3 of 4)

- Bentuk komplek dari Inverted Index :

Term	Inverted List
Algoritma	<Id1,1,[1]>, <id3,2,[1,4]>
Dapat	<Id1,1,[3]>
Digunakan	<Id1,1,[4]>
Fungsi	<Id2,1,[2]>
Fuzzy	<Id1,1,[7]>, <id2,1,[5]>
Genetik	<Id1,1,[2]>, <id3,1,[2]>
Keanggotaan	<Id2,1,[3]>
Learning	<Id3,1,[5]>
Merupakan	<Id3,1,[3]>
Optimasi	<Id1,1,[6]>, <id2,1,[1]>
Pada	<Id2,1,[4]>
Untuk	<Id1,1,[5]>

- Kemudian misalkan mencari hasil Boolean Query Retrieval : **Fuzzy OR NOT (Genetik AND Learning)**

Case Study B (4 of 4)

- Kemudian misalkan mencari hasil Boolean Query Retrieval : **Fuzzy OR NOT (Genetik AND Learning)**
  - TF_{biner}(Fuzzy) = 110
  - TF_{biner}(Genetik) = 101
  - TF_{biner}(Learning) = 001
- **Fuzzy OR NOT (Genetik AND Learning)**
  = 110 OR NOT (101 AND 001)
  = 110 OR NOT (001)
  = 110 OR 110
  = **110**

Term	Inverted List
Algoritma	<id1,1,[1]>, <id3,2,[1,4]>
Dapat	<id1,1,[3]>
Digunakan	<id1,1,[4]>
Fungsi	<id2,1,[2]>
Fuzzy	<id1,1,[7]>, <id2,1,[5]>
Genetik	<id1,1,[2]>, <id3,1,[2]>
Keanggotaan	<id2,1,[3]>
Learning	<id3,1,[5]>
Merupakan	<id3,1,[3]>
Optimasi	<id1,1,[6]>, <id2,1,[1]>
Pada	<id2,1,[4]>
Untuk	<id1,1,[5]>

- Jadi hasil Boolean Query Retrieval : **Fuzzy OR NOT (Genetik AND Learning)** adalah Dokumen “1 dan 2”.

Selesai