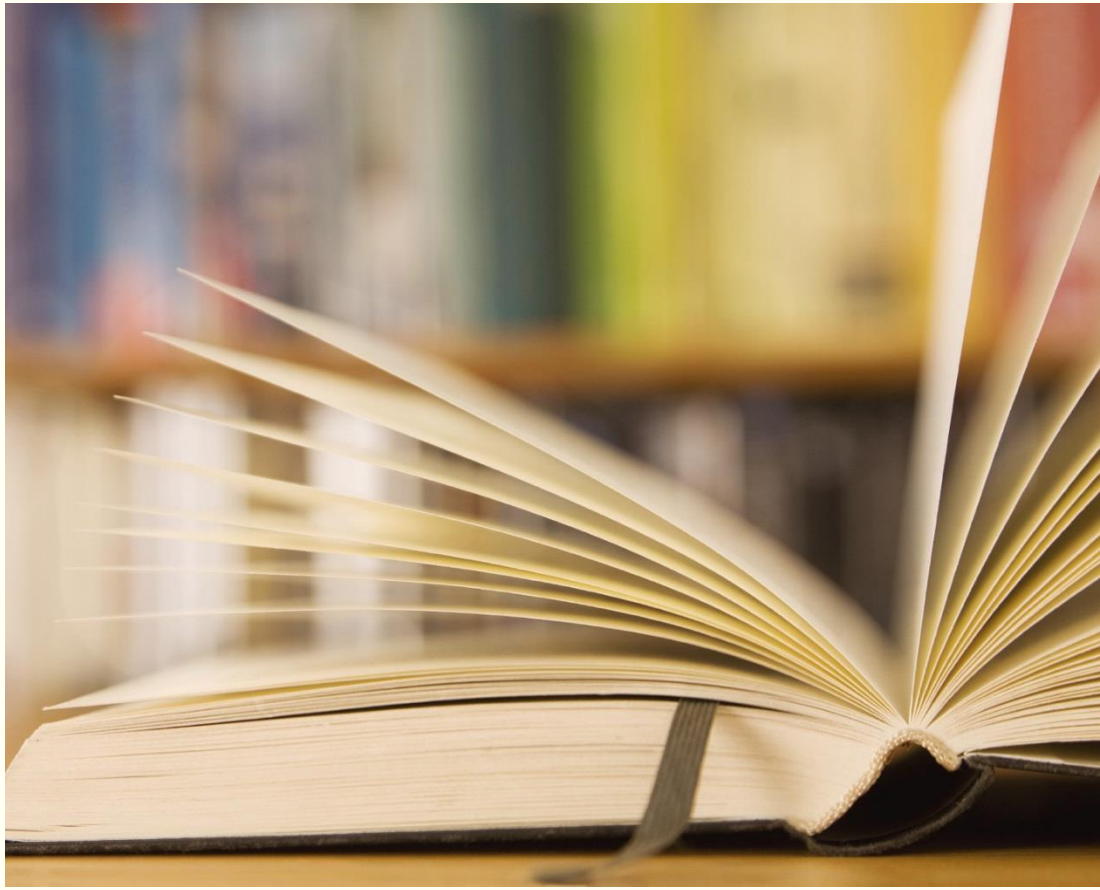
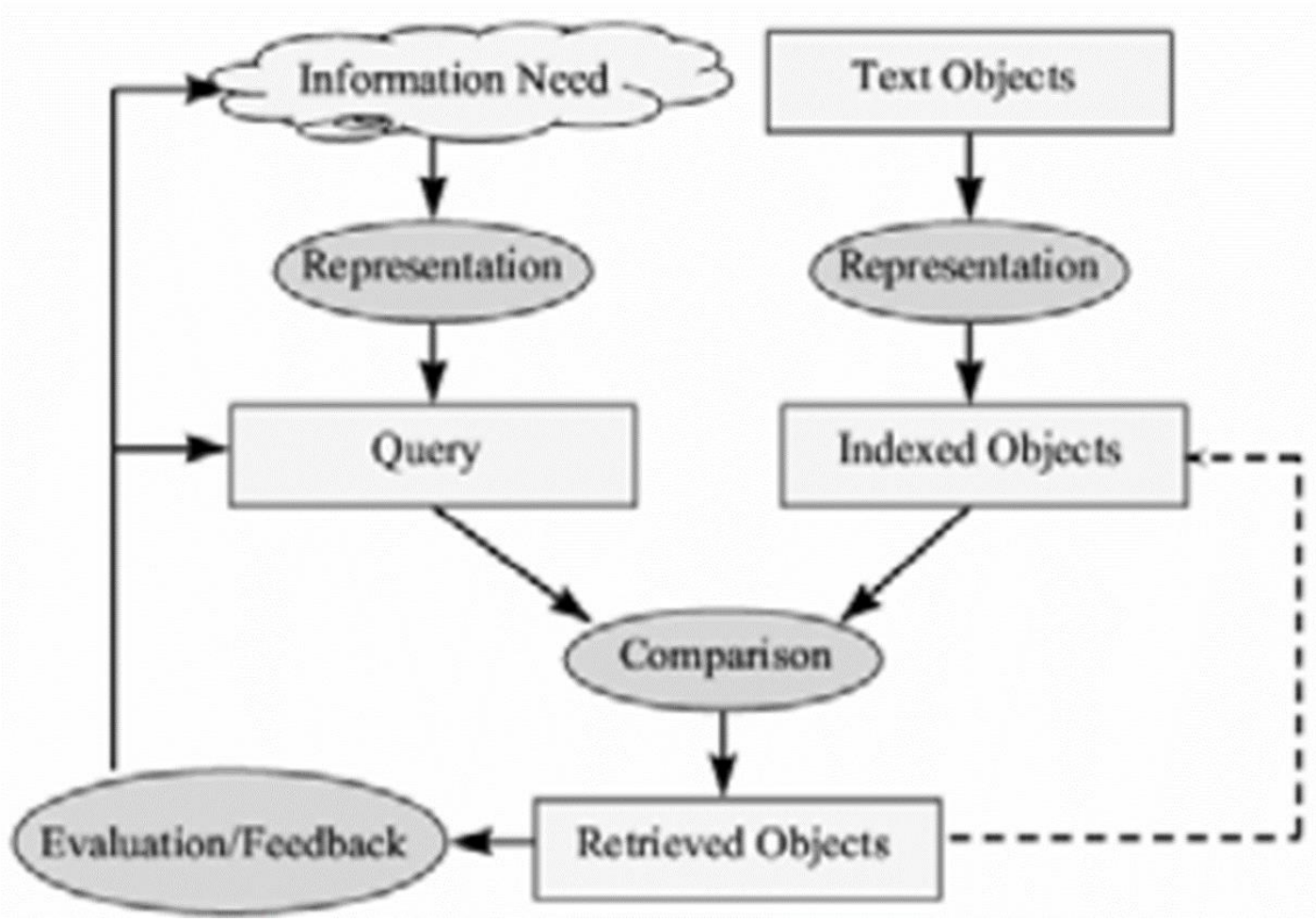
PEMODELAN IR

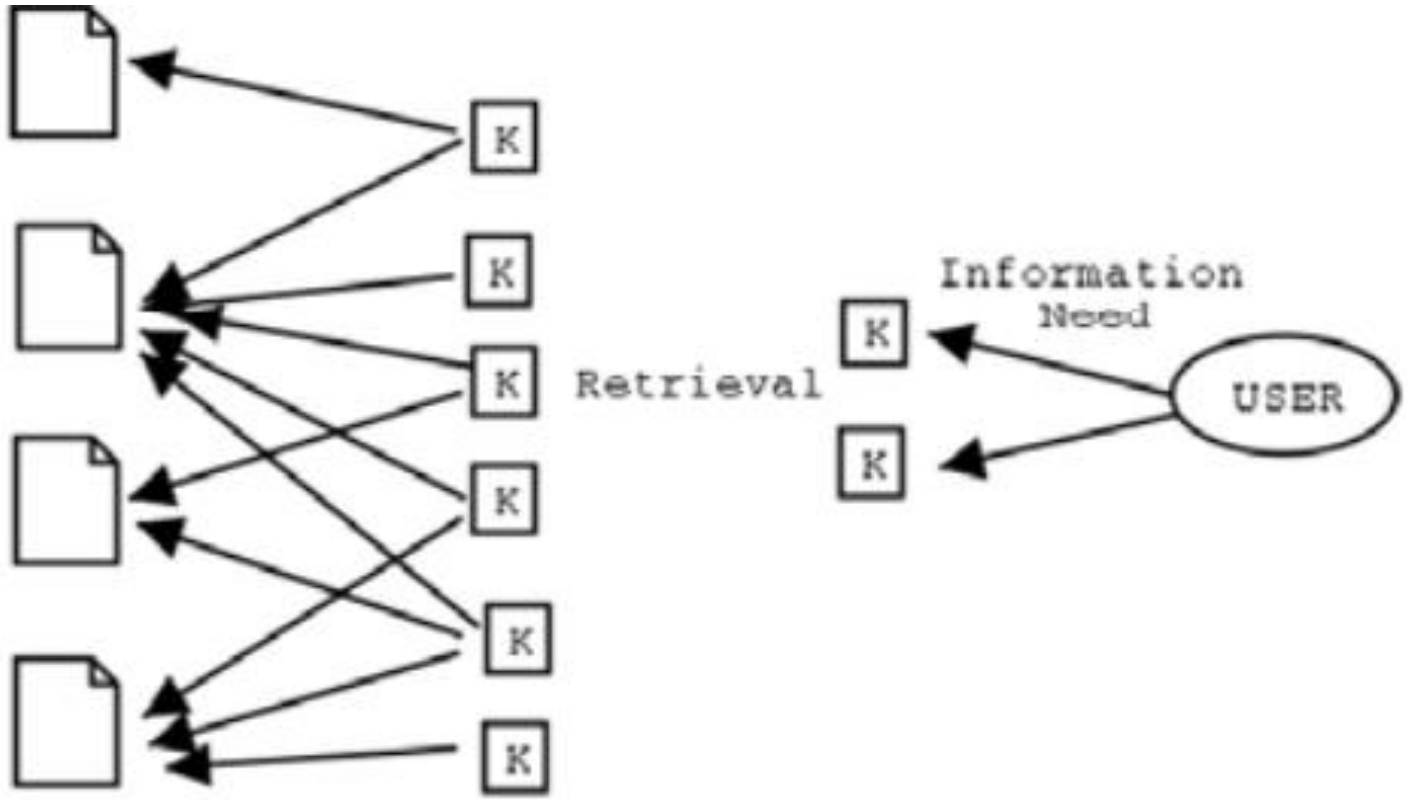
Oleh : Rahmat Robi Waliyansyah, M.Kom.



PROSES TEMU-KEMBALI



Konsep IR

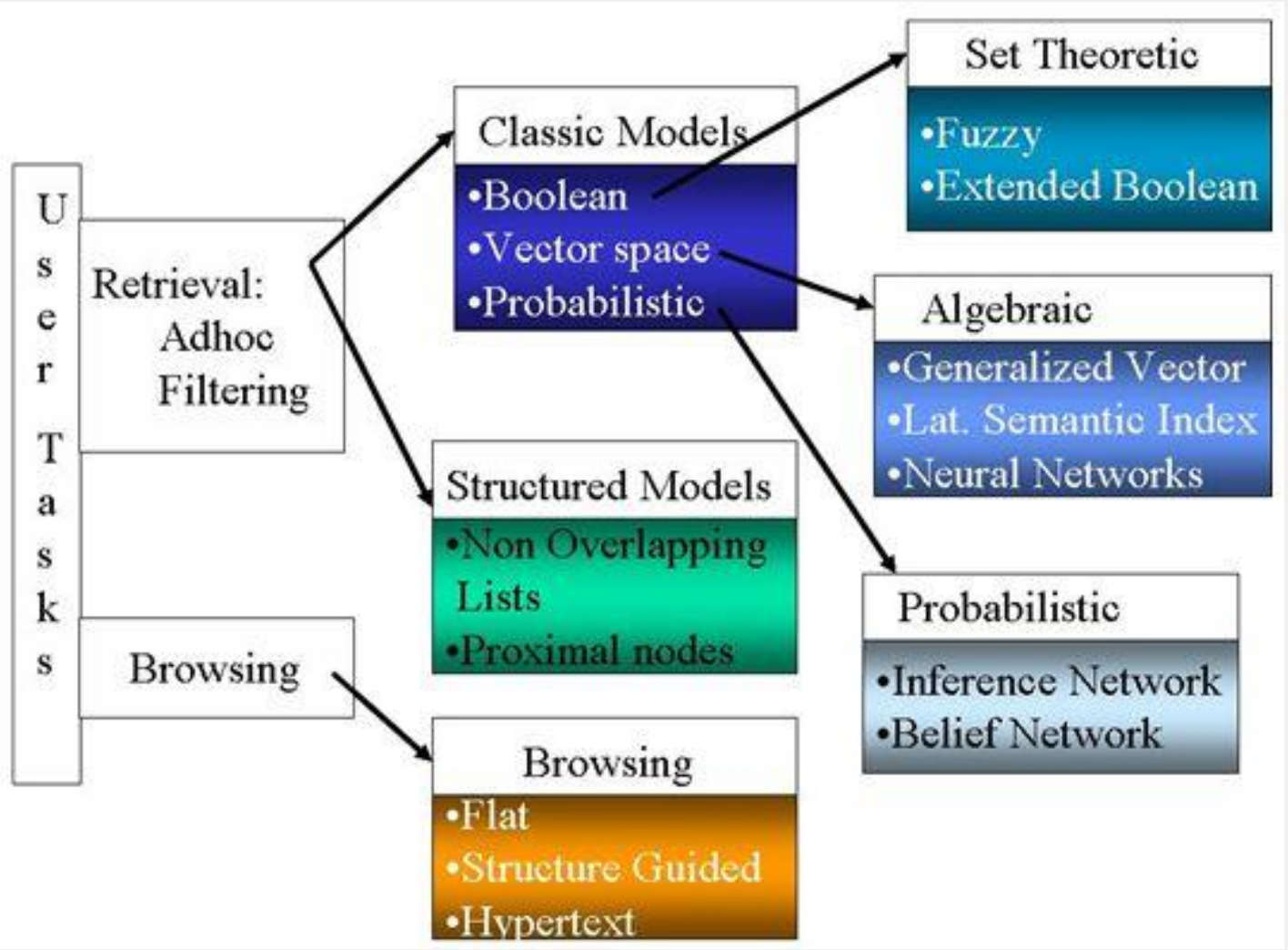


Pemodelan IR

- Model IR didefinisikan sebagai empat komponen [D, F, Q, R(q, dj)]
- Keterangan:

  - D adalah kumpulan dokumen
  - Q adalah query
  - F menunjukkan pemodelan dokumen dan query
  - R(q, dj) adalah fungsi peringkat yang dikaitkan dengan suatu nilai €R, dimana q€Q dan dj€D

Model IR


User Tasks
Retrieval: Adhoc Filtering
Browsing
Classic Models
•Boolean
•Vector space
•Probabilistic
Structured Models
•Non Overlapping Lists
•Proximal nodes
Browsing
•Flat
•Structure Guided
•Hypertext
Set Theoretic
•Fuzzy
•Extended Boolean
Algebraic
•Generalized Vector
•Lat. Semantic Index
•Neural Networks
Probabilistic
•Inference Network
•Belief Network

Boolean Model

- Exact match, pencocokan secara tepat sama.
- Query berbentuk ekspresi boolean.
- Dokumen bisa cocok atau tidak cocok dengan query yang diberikan.
- Hasilnya berupa sekumpulan dokumen yang cocok.
- Tidak ada peringkat dokumen sesuai dengan query yang diberikan.

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*"

Boolean Model

Keuntungan

- Implementasi mudah dan sederhana.
- Query mudah disusun dan dimengerti.
- Operator AND, OR, NOT sesuai dengan bahasa alami.

Kelemahan

- Pencocokan yang tepat dapat mengambil dokumen terlalu sedikit atau terlalu banyak.
- Sulit untuk menerjemahkan query ke dalam ekspresi Boolean.
- Semua istilah sama-sama berbobot.