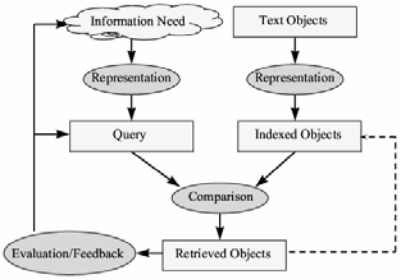
# KOM341
# Temu Kembali Informasi

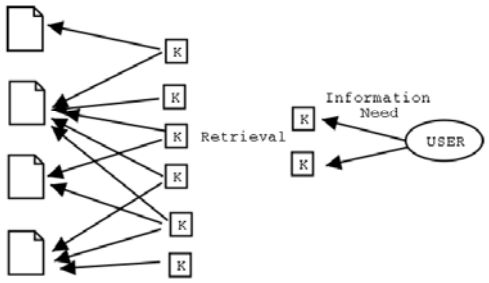
KULIAH #4
- Pemodelan IR
- Boolean model
- Vector space model

## Proses Temu-Kembali






## Konsep IR



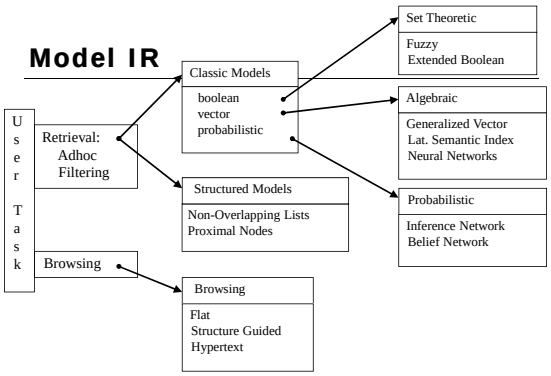



## Pemodelan IR

- Model IR didefinisikan sebagai empat komponen [D, F, Q, R(q, $d_j$)]
- Keterangan:
  - D adalah kumpulan dokumen
  - Q adalah query
  - F menunjukkan pemodelan dokumen dan query
  - R(q, $d_j$) adalah fungsi peringkat yang dikaitkan dengan suatu nilai $\in$R, dimana q$\in$Q dan $d_j \in$D



## Model IR






## 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 Model

- Bobot $w_{t,d} \in \{0,1\}$
- Query ***q*** terdiri dari kata, frase, atau konsep yang dihubungkan dengan operator Boolean AND, OR, atau NOT.
- Contoh:
  ***q*** = $[k_a \wedge (k_b \vee \neg k_c)]$ = $k_a$ && ($k_b$ || !$k_c$)



## Contoh

d1 ➔ And the **angel**s, all **pallid** and **wan**,
d2 ➔ **Upris**ing, **unveil**ing, **affirm**
d3 ➔ That the **play** is the **tragedy**, "**Man**,"
d4 ➔ **Angel** and its **hero** the **Conqueror Worm**.

Hasil Tokenisasi:

| | |
|---|---|
| 1) affirm | 7) play |
| 2) angel | 8) tragedy |
| 3) conqueror | 9) unveil |
| 4) hero | 10) uprise |
| 5) man | 11) wan |
| 6) pallid | 12) worm |



## Pembobotan Boolean

Contoh query:
**hero** AND (**angel** OR NOT **man**)

Formulasi query :
= $[k_4 \wedge \{k_2 \vee \neg k_5\}]$
= [(0 1 0 1) ∧ {(1 0 0 0} ∨ ¬ (0 0 1 0)}]
= (0 1 0 1)

Hasil query (tidak ada urutan):
$d_2$ dan $d_4$

| | d1 | d2 | d3 | d4 |
|---|---|---|---|---|
| affirm | 0 | 1 | 0 | 0 |
| angel | 1 | 0 | 0 | 0 |
| conqueror | 0 | 0 | 0 | 1 |
| hero | 0 | 1 | 0 | 1 |
| man | 0 | 0 | 1 | 0 |
| pallid | 1 | 0 | 0 | 0 |
| play | 0 | 0 | 1 | 0 |
| tragedy | 0 | 0 | 1 | 0 |
| unveil | 0 | 1 | 0 | 0 |
| uprise | 0 | 1 | 0 | 0 |
| wan | 1 | 0 | 0 | 0 |
| worm | 0 | 0 | 0 | 1 |



## Boolean Model

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

Kelemahan
- Tidak ada peringkat dokumen sesuai dengan query yang diberikan
- Exact matching
- Repot untuk query yang kompleks



## Boolean Scoring : Linear zone combinations

- Contoh:
  tiap dokumen memiliki dua zona, yaitu title dan body (atau text).
- Untuk setiap w∈[0,1] dapat dihitung:

$$score(d,q)=w.s_T(d,q) + (1-w).s_B(d,q)$$

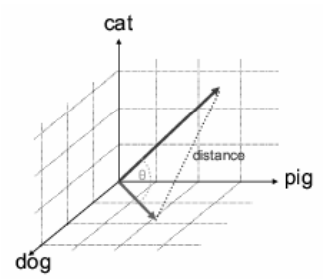
$s_T(d, q) \in \{0,1\}$ : nilai Boolean q dalam Title
$s_B(d, q) \in \{0,1\}$ : nilai Boolean q dalam Body



## Vector Space Model

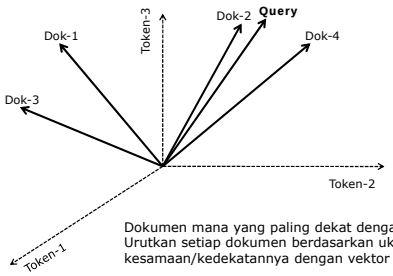
- Model berbasis token
- Memungkinkan partial matching dan pemeringkatan dokumen. Cenderung sebagai best matching.
- Prinsip dasar:
  - Dokumen sebagai vektor token
  - Terdapat ***t*** kumpulan token
  - Query sebagai vektor token
  - Kesamaan vektor dokumen dan query dihitung berdasarkan jarak atau kesamaan antar vektor

## Model Geometrik





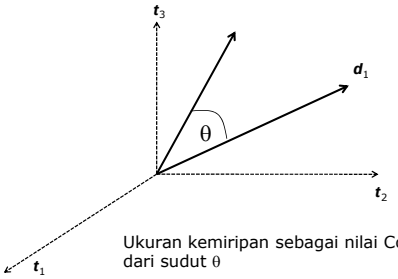

## Kesamaan Antar Vektor




Dokumen mana yang paling dekat dengan query? Urutkan setiap dokumen berdasarkan ukuran kesamaan/kedekatannya dengan vektor query



## Ukuran kemiripan Cosine




Ukuran kemiripan sebagai nilai Cosinus dari sudut θ



## Ukuran kemiripan Cosine

- Ukuran kesamaan Cosine antara $\boldsymbol{d}_j$ dan $\boldsymbol{d}_k$

$$sim(d_j, d_k) = \frac{\vec{d}_j \bullet \vec{d}_k}{\left\|\vec{d}_j\right\| \times \left\|\vec{d}_k\right\|}$$

- Panjang vektor

$$\left\|\vec{d}\right\| = \sqrt{\vec{d}\,'.\vec{d}}$$



## Nilai koefisien vektor

- Koefisien vektor menunjukkan seberapa penting suatu kata
- VSM tidak memberi ketentuan mengenai nilai koefisien vektor (bobot kata)
- Beberapa contoh nilai bobot
  - {0, 1}
  - tf
  - tf.idf



| | |
|---|---|
| Query | Petani mengalami gagal panen. |
| D1 | Gagal panen banyak yang terjadi. |
| D2 | Panen raya banyak dilaksanakan. |
| D3 | Jalan raya sering terjadi kecelakaan. |
| D4 | Petani gagal tanam karena mengalami panen yang gagal. |

| Kata | tf | | | | | N/n | idf | Bobot (w) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q | D1 | D2 | D3 | D4 | | | Q | D1 | D2 | D3 | D4 |
| banyak | 0 | 1 | 1 | 0 | 0 | 4/2=2 | 0.301 | 0 | 0.301 | 0.301 | 0 | 0 |
| dilaksanakan | 0 | 0 | 1 | 0 | 0 | 4/1=4 | 0.602 | 0 | 0 | 0.602 | 0 | 0 |
| gagal | 1 | 1 | 0 | 0 | 2 | 4/2=2 | 0.301 | 0.301 | 0.301 | 0 | 0 | 0.602 |
| jalan | 0 | 0 | 0 | 1 | 0 | 4/1=4 | 0.602 | 0 | 0 | 0 | 0.602 | 0 |
| karena | 0 | 0 | 0 | 0 | 1 | 4/1=4 | 0.602 | 0 | 0 | 0 | 0 | 0.602 |
| kecelakaan | 0 | 0 | 0 | 1 | 0 | 4/1=4 | 0.602 | 0 | 0 | 0 | 0.602 | 0 |
| mengalami | 1 | 0 | 0 | 0 | 1 | 4/1=4 | 0.602 | 0.602 | 0 | 0 | 0 | 0.602 |
| panen | 1 | 1 | 1 | 0 | 1 | 4/3=1.3 | 0.125 | 0.125 | 0.125 | 0.125 | 0.000 | 0.125 |
| petani | 1 | 0 | 0 | 0 | 1 | 4/1=4 | 0.602 | 0.602 | 0 | 0 | 0 | 0.602 |
| raya | 0 | 0 | 1 | 1 | 0 | 4/2=2 | 0.301 | 0 | 0 | 0.301 | 0.301 | 0 |
| sering | 0 | 0 | 0 | 1 | 0 | 4/1=4 | 0.602 | 0 | 0 | 0 | 0.602 | 0 |
| tanam | 0 | 0 | 0 | 0 | 1 | 4/1=4 | 0.602 | 0 | 0 | 0 | 0 | 0.602 |
| terjadi | 0 | 1 | 0 | 1 | 0 | 4/2=2 | 0.301 | 0 | 0.301 | 0 | 0.301 | 0 |
| yang | 0 | 1 | 0 | 0 | 1 | 4/2=2 | 0.301 | 0 | 0.301 | 0 | 0 | 0.301 |

## Ukuran kemiripan Dot Product

Dot product vektor $\boldsymbol{d}_j$ dan $\boldsymbol{q}$

$$sim(d_j, q) = \vec{d}_j \bullet \vec{q}$$

- sim(D1, Q) = 0.106
- sim(D2, Q) = 0.016
- sim(D3, Q) = 0.000
- sim(D4, Q) = 0.922

| Urutan: |
|---|
| D4 |
| D1 |
| D2 |
| D3 |



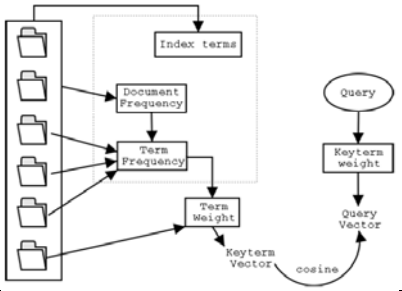
## Ukuran kemiripan Cosine

- Panjang vektor
  - |Q| = 0.912
  - |D1| = 0.615
  - |D2| = 0.748
  - |D3| = 1.126
  - |D4| = 1.385
- Ukuran kesamaan Cosine
  - sim(D1, Q) = 0.189
  - sim(D2, Q) = 0.023
  - sim(D3, Q) = 0.000
  - sim(D4, Q) = 0.730

| Urutan: |
|---|
| D4 |
| D1 |
| D2 |
| D3 |



## Prosedur






## Masalah komputasi

- Jika ukuran koleksi = N sangat besar (jutaan, milyaran, …), berapa nilai kompleksitas untuk menentukan urutan dokumen dari satu query pada N dokumen pada koleksi?
- Sangat besar sehingga waktu komputasi akan sangat lama.
- Cluster pruning : preprocessing untuk mengelompokkan dokumen dalam koleksi sesuai dengan kedekatan vektor.

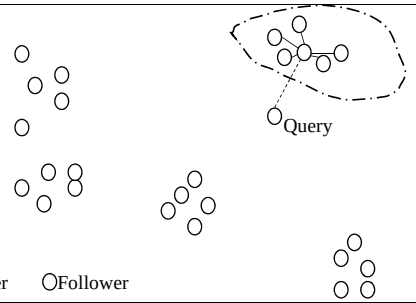


## Cluster pruning

- Prosedur (preprocessing):
  - Ambil secara acak √N dokumen. Disebut sebagai leaders.
  - Untuk setiap dokumen yang bukan leader (disebut followers), hitung kedekatannya dengan leader.
- Proses query q:
  - Dapatkan leader L yang dekat dengan q.
  - Cari K dokumen terdekat q di antara follower dari L



## Visualisasi Cluster pruning



## Latihan
## Gunakan tf.idf dan Cosine

- Dokumen:
  - $d_1$: "**Shipment** of **gold damaged** in a **fire**"
  - $d_2$: "**Delivery** of **silver arrived** in a **silver truck**"
  - $d_3$: "**Shipment** of **gold arrived** in a **truck**"

- Query: "gold silver truck "
- Asumsi : N=1000

| | arrived | damaged | delivery | fire | gold | shipment | silver | truck |
|---|---|---|---|---|---|---|---|---|
| df | 3 | 7 | 10 | 8 | 5 | 15 | 12 | 5 |

JULIO ADISANTOSO - ILKOM IPB