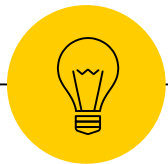


# Sistem Temu Kembali Informasi

“Model Sistem Temu Kembali Informasi (Vector Space Model)”





# Course **Outline**

1. Pengantar Information Retrieval
2. Dokumen Preprocessing
- 3. Model Information Retrieval**
4. Term Weighting

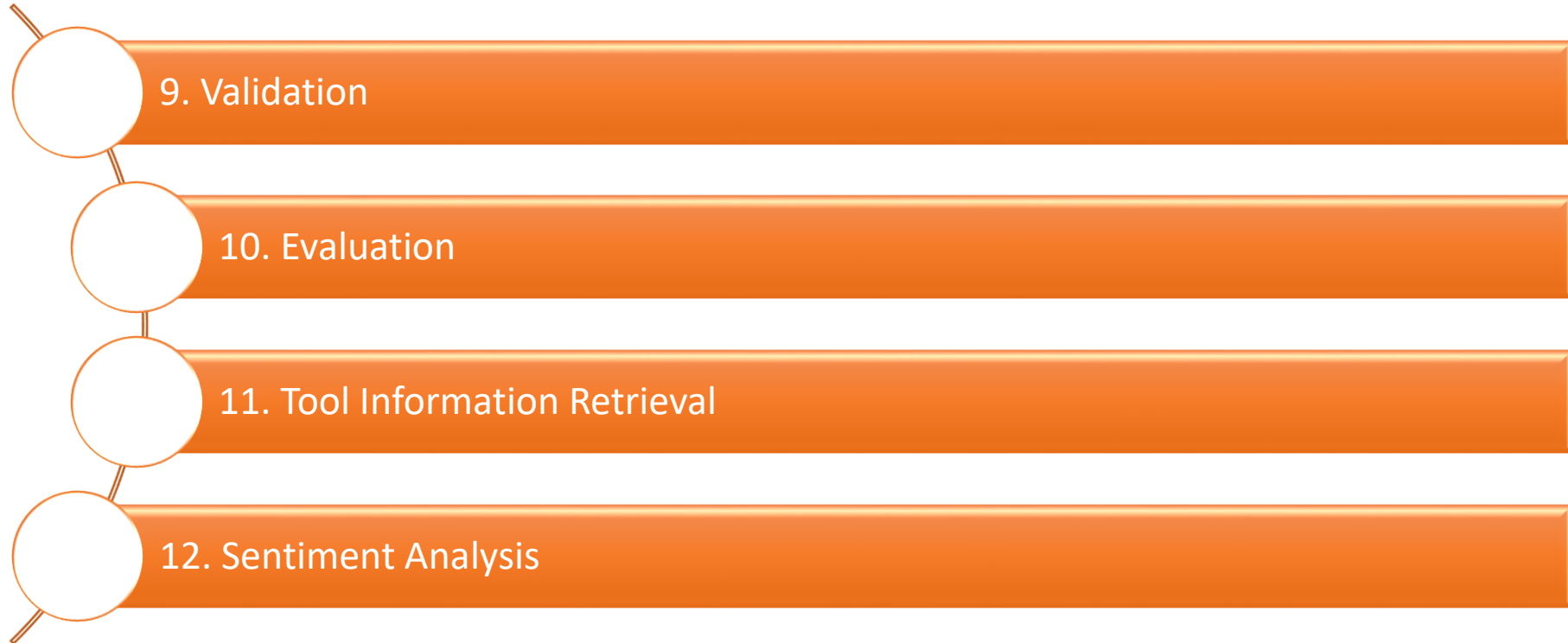


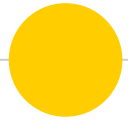
## Course Outline



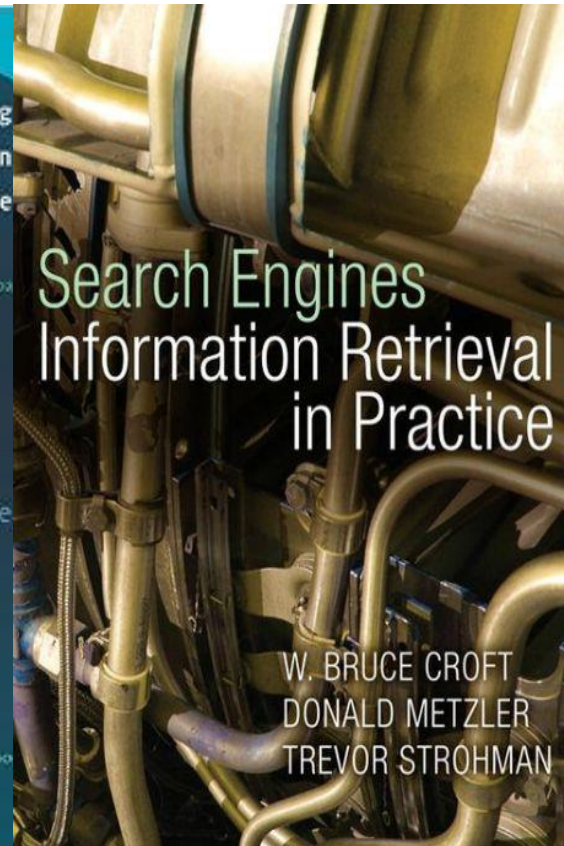
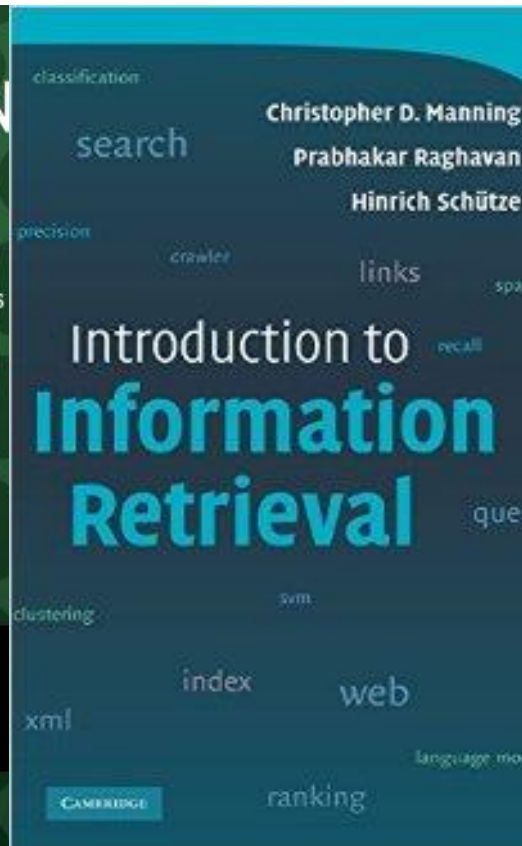
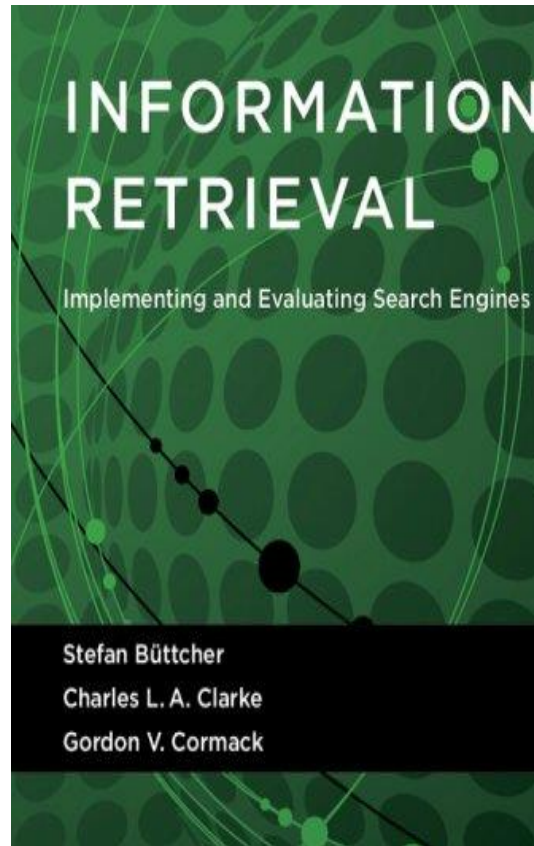


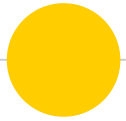
## Course **Outline**





## Buku Penunjang

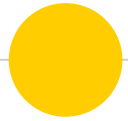




## Boolean Model

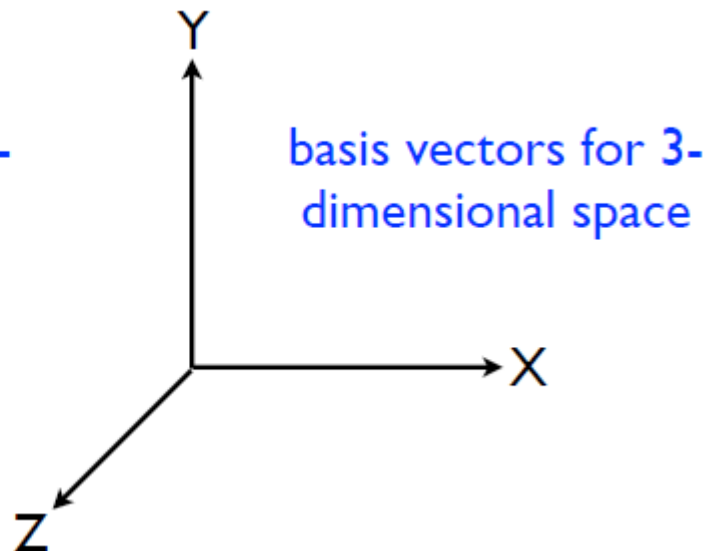
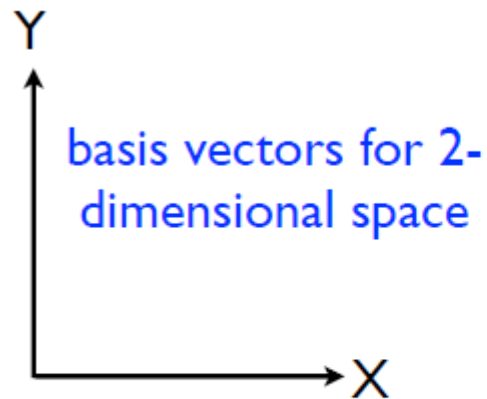
- Setiap 'term' hanya disimbolkan dengan
  - 1 : Muncul dalam dokumen
  - 0 : Tidak muncul dalam dokumen
- Belum merepresentasikan informasi tentang frekuensi kemunculan kata, lokasi kata, atau urutan kemunculan kata.

	<i>a</i>	<i>aardvark</i>	<i>abacus</i>	<i>abba</i>	<i>able</i>	<i>...</i>	<i>zoom</i>
<i>doc_1</i>	1	0	0	0	0	<i>...</i>	1
<i>doc_2</i>	0	0	0	0	1	<i>...</i>	1
<i>::</i>	<i>::</i>	<i>::</i>	<i>::</i>	<i>::</i>	<i>::</i>	<i>...</i>	0
<i>doc_m</i>	0	0	1	1	0	<i>...</i>	0

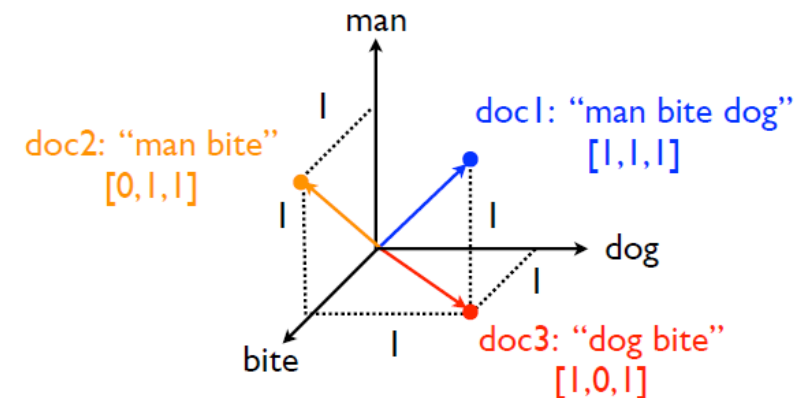


## Vector Space Model

- Memodelkan 'term' kedalam ruang vektor.
- Setiap titik memiliki jarak dan arah di dalam ruang vektor



- Setiap kata (term) merupakan satu dimensi dalam ruang vektor.
- Setiap titik merupakan satu dokumen.
- Nilai vektor merepresentasikan jumlah kemunculan kata tertentu dalam dokumen tersebut



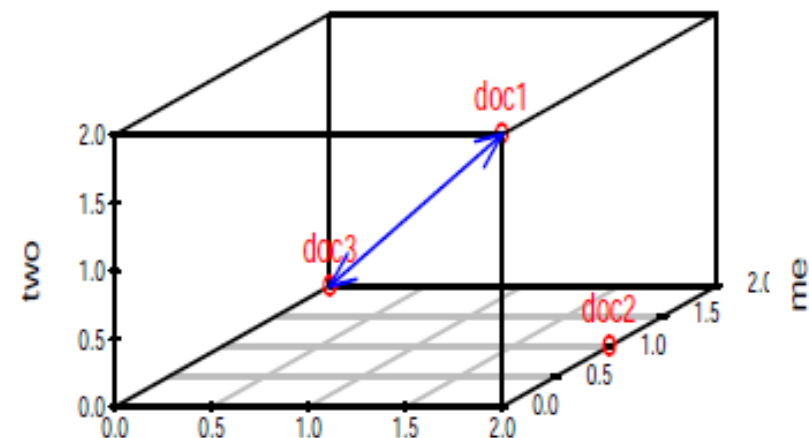


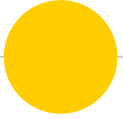


## Similarity

- Kemiripan dua vektor bisa dihitung dari seberapa dekat dua titik tersebut dalam ruang vektor.
- Salah satunya dengan menghitung jarak dengan euclidean distance.

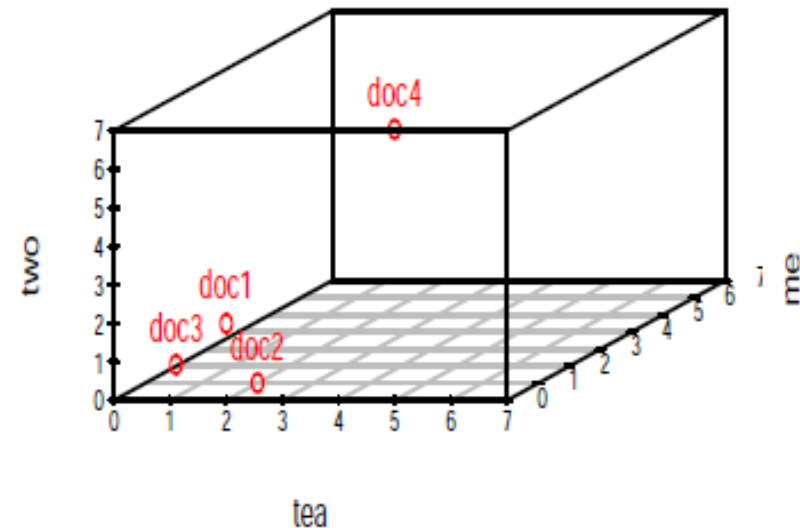
$$\begin{aligned} \text{sim}(\text{doc1}, \text{doc3}) \\ &= -\sqrt{2^2 + 2^2 + 2^2} \\ &= -\sqrt{12} \end{aligned}$$





- Memiliki kelemahan karena hanya menghitung kemiripan dari jumlah kemunculan kata.

Point	tea	me	two
doc1	2	0	2
doc2	2	1	0
doc3	0	2	0
doc4	5	0	7

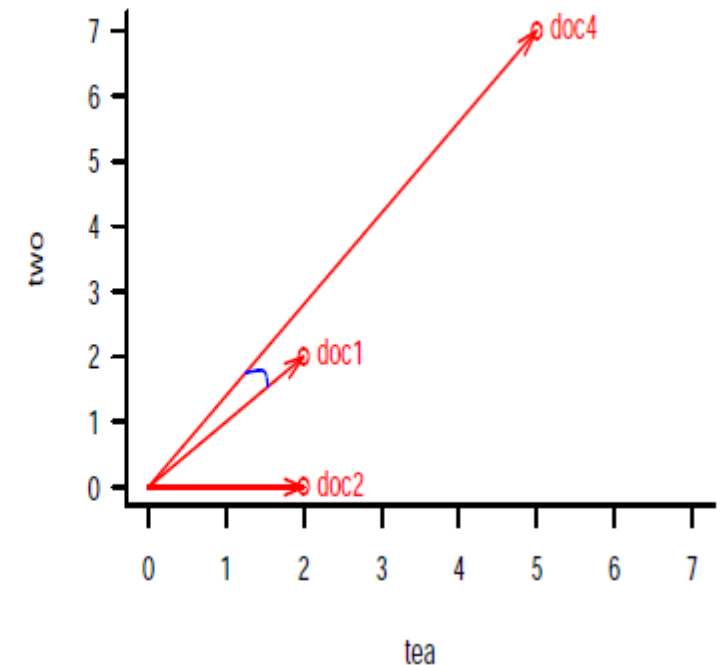





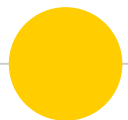
## Angular Similarity

- Contoh penghitungan kemiripan lain adalah berdasarkan sudut diantara 2 vektor.

Point	tea	two
doc1	2	2
doc2	2	0
doc4	5	7



- 
- 
- Contoh perhitungan kemiripan dengan menghitung nilai  $\cos$  dari sudut diantara dua vektor.



## Similarity

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

$$\vec{x} = (1, 1, 1, 3, 0)$$

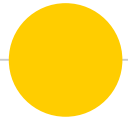
$$\vec{y} = (0, 0, 0, 1, 1)$$

$$\vec{x} \cdot \vec{y} = 0 + 0 + 0 + 3 + 0 = 3$$

$$|\vec{x}| = \sqrt{1 + 1 + 1 + 9 + 0} = \sqrt{12}$$

$$|\vec{y}| = \sqrt{0 + 0 + 0 + 1 + 1} = \sqrt{2}$$

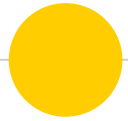
$$\cos(\vec{x}, \vec{y}) = \frac{3}{\sqrt{12} \times \sqrt{2}} = \frac{3}{\sqrt{24}} = 0.61$$



## Document Ranking

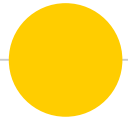
---

- Misalkan  $q$  sebagai vector dari query dan  $(D1, D2, \dots, Dn)$  sebagai dokumen yang akan dibandingkan.
- Hitung nilai kemiripan kosinus  $q$  dengan setiap dokumen  $(D1, D2, \dots, Dn)$
- Urutkan dokumen berdasarkan nilai kemiripan yang dihasilkan.



## Pengertian Vector Space Model (VSM) - (1)

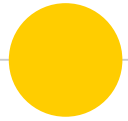
- *Vector Space Model (VSM)* adalah model *Information Retrieval* yang berbasis token untuk memungkinkan *partial matching* dan pemeringkatan dokumen (*pengindexan*).
- Atau model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*.
- Dengan prinsip dasar mengubah suatu dokumen menjadi kumpulan token-token.
- *Query* menjadi *vector* token yang berfungsi untuk mencari token yang berhubungan dengan melihat kesamaan *vector* dokumen dan *query* berdasarkan jarak *vector*.



## Pengertian Vector Space Model (VSM) - (2)

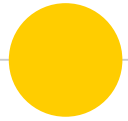
- ◎ *Query* dan dokumen dianggap sebagai vector-vector pada ruang  $n$ -dimensi  $d_i = (t_{i,1}, t_{i,2}, \dots, t_{i,k})$ , dimana  $t$  adalah jumlah dari seluruh *term* yang ada dalam leksikon.
- ◎ Leksikon adalah daftar semua term yang ada dalam indeks.
- ◎ Untuk setiap *term* tidak hanya dilihat ada atau tidak ada dalam dokumen namun diberi bobot menggunakan *scheme TF* atau *TF-IDF*.
- ◎ *TF-IDF* merupakan skema pembobotan yang sering digunakan dalam *Vector Space Model (VSM)* bersama dengan *cosine similarity* untuk menentukan kesamaan antara dua buah dokumen.





## Pembobotan Kata (TF-IDF) – (1)

- TF –IDF mempertimbangkan frekuensi kata-kata yang berbeda dalam semua dokumen dan mampu membedakan dokumen. Dalam *Vector Space Model (VSM)*, setiap vector disusun oleh term dan bobot yang mewakili dokumen.
- Kesamaan dokumen dapat dinyatakan dengan sudut atau jarak antara vector, semakin kecil sudut atau jarak berarti semakin mirip dua dokumen tersebut.



## Pembobotan Kata (TF-IDF) – (2)

- *TF (Term Frequency)* merupakan frekuensi kemunculan *term (t)* pada dokumen (d).
- *DF (Document Frequency)* merupakan banyaknya dokumen dimana suatu *term (t)* muncul. Atau sebuah dokumen yang mengandung *term (t)*.
- *IDF (Inverse Document Frequency)*

## Pembobotan Kata (TF-IDF) – (3)

☉ Rumus dari *TF-IDF* adalah sebagai berikut :

$$W_{t,d} = TF_{t,d} * IDF_t \quad (1)$$

Keterangan :

$W_{t,d}$  = bobot dari  $t$  (*term*) dalam satu dokumen

$TF_{t,d}$  = frekuensi kemunculan  $t$  (*term*) dalam dokumen  $d$

$IDF_t$  = *Inverse document frequency*, dimana

$$IDF_t = \log\left(\frac{N}{n_t}\right) \quad (2)$$

Keterangan :

$N$  = jumlah semua dokumen

$n_t$  = jumlah dokumen yang mengandung *term*  $t$



## Perankingan

- Prangkingan dilakukan untuk mengukur kemiripan antara *vector query* dan *vector document* yang akan dibandingkan.
- Salah satu metode yang biasa digunakan dalam perhitungan kemiripan adalah pengukuran *cosine*, yang menentukan sudut antara *vector document* dan *vector query* yang didefinisikan sebagai :

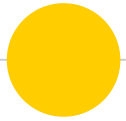
$$\text{Similarity}(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2 \cdot \sum_{i=1}^t w_{iq}^2}} \quad (3)$$

- Dimana,  $w_{q,t}$  adalah bobot dari *term* ( $t$ ), penyebut dalam persamaan ini disebut faktor normalisasi yang berfungsi untuk menghilangkan pengaruh panjang dokumen.
- Panjang dokumen cenderung memiliki nilai yang lebih besar karena memiliki frekuensi kemunculan kata yang besar pula.



## Cosine Similarity – (1)

- Fungsi *similarity* adalah fungsi yang menerima dua buah objek dan mengembalikan nilai kemiripan (*similarity*) antara kedua objek tersebut berupa bilangan riil.
- Umumnya, nilai yang dihasilkan oleh fungsi *similarity* berkisar pada interval  $[0...1]$ . Namun, ada juga yang menghasilkan nilai yang berada diluar interval tersebut. Maka untuk memetakannya dapat dilakukan normalisasi.
- *Cosine similarity* adalah perhitungan kesamaan antara dua *vector*  $n$  dimensi dengan mencari kosinus dari sudut diantara keduanya dan sering digunakan untuk membandingkan dokumen dalam *text mining*.



## Cosine Similarity – (2)

● Berikut rumus Cosine Similarity :

$$\text{Similarity}(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|} \quad (4)$$

Dimana :

$x \cdot y$  : *vector dot product* dari  $x$  dan  $y$ , dihitung dengan  $\sum_{k=1}^n x_k y_k$  (5)

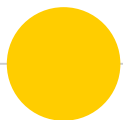
$\|x\|$  : panjang vektor  $x$ , dihitung dengan  $\sum_{k=1}^n x_k^2$  (6)

$\|y\|$  : panjang vektor  $y$ , dihitung dengan  $\sum_{k=1}^n y_k^2$  (7)



## Cosine Similarity – (3)

- Semakin besar hasil fungsi similarity, maka kedua objek yang dievaluasi dianggap semakin mirip. Sebaliknya, jika semakin kecil hasil fungsi similarity, maka kedua objek tersebut dianggap semakin berbeda.
- Pada fungsi yang menghasilkan nilai pada jangkauan  $[0...1]$ , nilai 1 melambangkan kedua objek persis sama, sedangkan nilai 0 melambangkan kedua objek sama sekali berbeda.



## Contoh Perhitungan - (1)

Data yang digunakan dalam penelitian ini terdiri dari 12 hama dan 11 penyakit tanaman padi dengan total jumlah gejala sebanyak 48 dalam 3 fase tanam. Untuk proses identifikasi *input*, 48 gejala yang ada akan melalui tahap *preprocessing* untuk dihitung kemiripan dengan *query*. Gejala yang ada direpresentasikan sebagai dokumen dengan kode G1, G2, ..., Gn dan *query user* juga akan dianggap sebagai dokumen pembanding dengan kode Q. Kemudian dilakukan tahap *preprocessing* sebagai berikut :

Menghapus format dan *markup* dalam gejala, kemudian dari tiap gejala dilakukan tokenisasi, yaitu memisahkan tiap kata dalam kalimat, selanjutnya dilakukan *filtering* untuk menghilangkan kata-kata sambung seperti “di”, “atau”, “seperti” dan lain-lain. Lalu tiap kata akan di-*stemming* dimana akan dihilangkan semua imbuhan yang melekat pada kata, seperti awalan dan akhiran.

Contoh :

G1 : Daun berwarna kuning

G2 : Daun berwarna putih

G3 : Batang berwarna kuning

Q : Daun kuning

Menjadi :

Daun, hijau, kuning, putih, dan batang.

Setelah didapatkan daftar *term*, kemudian dilakukan pembobotan dengan menghitung jumlah frekuensi kemunculan (tf) dan idf dengan rumus  $IDF_t = \log\left(\frac{N}{d_f}\right)$ ,

dimana N adalah jumlah keseluruhan gejala dan *query* dan df adalah jumlah gejala dan *query* yang memiliki *term* t.



## Contoh Perhitungan – (2)

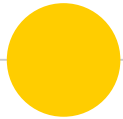
Tabel 1. Perhitungan tf dan idf

<i>Term</i>	<b>Q</b>	<b>G1</b>	<b>G2</b>	<b>G3</b>	<b>IDF <math>\left(\log \frac{N}{df}\right)</math></b>
Batang	0	0	0	1	$\log \left(\frac{4}{1}\right) = 0.60$ 206
Daun	1	1	1	0	$\log \left(\frac{4}{3}\right) = 0.12$ 494
Kuning	1	1	0	1	$\log \left(\frac{4}{3}\right) = 0.12$ 494
Putih	0	0	1	0	$\log \left(\frac{4}{1}\right) = 0.60$ 206
Warna	0	1	1	1	$\log \left(\frac{4}{3}\right) = 0.12$ 494

Kemudian dihitung bobotnya menggunakan rumus  $TF \times IDF$  :

Tabel 2. Perhitungan pembobotan

<i>Term</i>	<b>Q</b>	<b>G1</b>	<b>G2</b>	<b>G3</b>
Batang	0	0	0	0.60206
Daun	0.12494	0.12494	0.12494	0
Kuning	0.12494	0.12494	0	0.12494
Putih	0	0	0.60206	0
Warna	0	0.12494	0.12494	0.12494



## Contoh Perhitungan – (3)

Kemudian dilakukan normalisasi sebagai berikut :

- Menghitung perkalian skalar antar bobot hitung dengan rumus  $\sum_{t=1}^v W_{q,t} * W_{i,t}$  dan panjang vektor  $\sqrt{\sum_{t=1}^v W_{q,t}^2}$

**Tabel 3. Hasil perhitungan panjang vektor**

<i>Term</i>	<b>Q</b>	<b>G1</b>	<b>G2</b>	<b>G3</b>
Batang	0	0	0	0.362476
Daun	0.01561	0.01561	0.01561	0
Kuning	0.01561	0.01561	0	0.01561
Putih	0	0	0.362476	0
Warna	0	0.01561	0.01561	0.01561
Jumlah	0.03122	0.04683	0.393696	0.393696
Akar	0.176692	0.216402	0.627452	0.627452

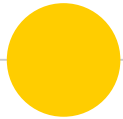
$$\text{Hasil cosine } (\vec{G_i}, \vec{q}) = \frac{\vec{G_i} \cdot \vec{q}}{|\vec{G_i}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2 \cdot \sum_{i=1}^t w_{iq}^2}}$$

$$G1 = \frac{0.03122}{0.176692 \cdot 0.216402} = \frac{0.03122}{0.038237} = 0,816497$$

$$G2 = \frac{0.01561}{0.176692 \cdot 0.627452} = \frac{0.01561}{0.110866} = 0,14080$$

$$G3 = \frac{0.01561}{0.176692 \cdot 0.627452} = \frac{0.01561}{0.110866} = 0,14080$$

Dari hasil ini didapatkan bahwa gejala yang paling mewakili maksud dari *input user* yang berupa gejala daun kuning adalah G1 yaitu gejala daun berwarna kuning dengan nilai kemiripan sebesar 0,816497.



## Contoh Perhitungan – (4)

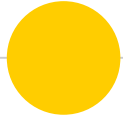
Hasil akhir perhitungan kemiripan antar *query* dan gejala akan dianggap sebagai *feedback*. Berikut ini adalah tahap-tahap identifikasi penyakit :

1. Melakukan transformasi biner, hasil perankingan dan penyakit yang ada ditransformasikan menjadi biner, dimana tiap gejala yang berupa hasil identifikasi *input* sebelumnya akan diberi nilai 1 dan yang lain diberi nilai biner 0.

**Tabel 5. Contoh transformasi biner untuk *feedback* serta hama wereng coklat dan hama putih palsu**

Hama atau penyakit	Gejala			
	Daun warna kuning	Daun berwarna putih	Daun berwarna hijau tua	Batang berwarna kuning
Hama wereng coklat	Ada	Tidak	Tidak	Ada
Hama wereng coklat (t)	1	0	0	1
Hama putih palsu	Tidak	Ada	Tidak	Tidak
Hama putih palsu (t)	0	1	0	0
<i>Feedback</i>	Ya	Tidak	Ya	Ya
<i>Feedback</i> (t)	1	0	1	1

**Ket: (t)** merupakan hasil tranformasi biner.



## Contoh Perhitungan – (5)

Menghitung nilai *cosine similarity* untuk deteksi penyakit.

Contoh :

$X = 1,0,0,1$  (hasil transformasi biner untuk hama wereng coklat)

$Y = 1,0,1,1$  (hasil transformasi biner *feedback*)

$$\begin{aligned} X.Y &= (1 * 1) + (0 * 0) + (0 * 1) + (1 * 1) \\ &= 1 + 0 + 0 + 1 = 2 \end{aligned}$$

$$\|x\| = \sqrt{1^2 + 0^2 + 0^2 + 1^2} = \sqrt{2} = 1,41421$$

$$\|y\| = \sqrt{1^2 + 0^2 + 1^2 + 1^2} = \sqrt{3} = 1,73205$$

$$\begin{aligned} \cos(x, y) &= \frac{x.y}{\|x\|.\|y\|} = \frac{2}{1,41421.1,73205} \\ &= \frac{2}{2,44729} = 0,816499 \end{aligned}$$

Sehingga didapat hasil *similarity* antara hama wereng coklat dan *feedback* adalah sebesar 0,816499.



## Kelebihan & Kekurangan Vector Space Model (VSM)

### Kelebihan

- ⦿ Adanya peringkat pengambilan informasi
- ⦿ Menampilkan referensi yang sesuai kebutuhan
- ⦿ Penyocokan secara *partial*

### Kekurangan

- ⦿ Menganggap informasi adalah independen
- ⦿ Bobot permasalahan (istilah) tidak lagi diperlukan



Thanks!

*Any questions ?*