Penambangan Data
Tugas 7
Alvaro Austin - 2106752180

1. Masyarakat bersiap menyambut Lebaran dengan penuh harap dan kegembiraan.
2. Indonesia menang 3 gol tanpa balas melawan Vietnam.
3. Korupsi timah merugikan negara Indonesia 271 Triliun
4. Libur lebaran akan dimulai pada tanggal 6 April 2024.
5. Perkembangan teknologi kecerdasan buatan sangat cepat
6. Saya makan nasi goreng untuk buka puasa
7. Indonesia berpeluang lolos ke piala dunia 2026

Soal

1. Lakukan tahap preprocessing untuk teks berita tersebut secara manual dan jelaskan setiap perubahan yang Anda lakukan dengan lengkap.
   Tahap 1: Lowercase texts and remove punctuation

   Output dari tahap ini adalah:
   1. masyarakat bersiap menyambut lebaran dengan penuh harap dan kegembiraan.
   2. indonesia menang 3 gol tanpa balas melawan vietnam
   3. korupsi timah merugikan negara indonesia 271 triliun
   4. libur lebaran akan dimulai pada tanggal 6 april 2024
   5. perkembangan teknologi kecerdasan buatan sangat cepat
   6. saya makan nasi goreng untuk buka puasa
   7. indonesia berpeluang lolos ke piala dunia 2026

   Tahap 2: Lemmatization (mengubah menjadi kata-kata dasar)
   1. bersiap -> siap, menyambut -> sambut, kegembiraan -> gembira
   2. melawan -> lawan
   3. merugikan -> rugi
   4. dimulai -> mulai
   5. perkembangan -> kembang, kecerdasan -> cerdas, buatan -> buat
   6. -
   7. berpeluang -> peluang

   Hasil akhir Lemmatization:
   1. masyarakat siap sambut lebaran dengan penuh harap dan gembira
   2. indonesia menang 3 gol tanpa balas lawan vietnam
   3. korupsi timah rugi negara Indonesia 271 triliun
   4. libur lebaran akan mulai pada tanggal 6 april 2024
   5. kembang teknologi cerdas buat sangat cepat
   6. saya makan nasi goreng untuk buka puasa
   7. indonesia peluang lolos ke piala dunia 2026

   Tahap 3: Stopwords removal (berdasarkan List Stop Word Indonesian)
   1. masyarakat ~~siap~~ sambut lebaran ~~dengan~~ penuh harap ~~dan~~ gembira
   2. indonesia menang 3 gol ~~tanpa~~ balas lawan vietnam
   3. korupsi timah rugi negara Indonesia 271 triliun
   4. libur lebaran ~~akan~~ ~~mulai~~ ~~pada~~ tanggal 6 april 2024
   5. kembang teknologi cerdas ~~buat~~ ~~sangat~~ cepat
   6. saya makan nasi goreng ~~untuk~~ buka puasa
   7. indonesia peluang lolos ~~ke~~ piala dunia 2026

   Hasil stopwords removal:
   1. masyarakat sambut lebaran penuh harap gembira
   2. indonesia menang 3 gol balas lawan vietnam
   3. korupsi timah rugi negara Indonesia 271 triliun
   4. libur lebaran tanggal 6 april 2024
   5. kembang teknologi cerdas cepat
   6. saya makan nasi goreng buka puasa
   7. indonesia peluang lolos piala dunia 2026

   Tahap 4 (terakhir): Bag of Words (memisahkan menjadi kata-kata)
   1. {masyarakat, sambut, lebaran, penuh, harap, gembira}
   2. {indonesia, menang, 3, gol, balas, lawan, vietnam}
   3. {korupsi, timah, rugi, negara, indonesia, 271, triliun}
   4. {libur, lebaran, tanggal, 6, april, 2024}
   5. {kembang, teknologi, cerdas, cepat}
   6. {saya, makan, nasi, goreng, buka, puasa}
   7. {indonesia, peluang, lolos, piala, dunia, 2026}

2. Lakukan vektorisasi unigram menggunakan algoritma TF-IDF secara manual dan jelaskan setiap langkah yang Anda lakukan dengan lengkap.

Semua token yang diperoleh dari nomor sebelumnya:
{3, rugi, menang, gembira, lebaran, tanggal, 6, kembang, teknologi, 2024, nasi, sambut, april, saya, gol, cerdas, penuh, triliun, lolos, goreng, balas, dunia, libur, puasa, piala, korupsi, 2026, lawan, cepat, harap, indonesia, vietnam, makan, peluang, 271, buka, negara, timah, masyarakat}

Keterangan Kolom:

- Token: adalah kata-kata yang diperoleh dari evaluasi hasil nomor 1 (bag of words) yang digabungkan
- Nt = Jumlah dokumen yang mengandung token tersebut.
- IDF = Jumlah dokumen dalam korpus (rumusnya sendiri adalah $log(\frac{N}{Nt}) = log(\frac{7}{Nt})$. N bernilai 7 karena ada 7 dokumen.
- TF = Jumlah kemunculan term pada dokumen tertentu
- TF-IDF = Perhitungan TF dikalikan dengan IDF token tertentu. Rumusnya: $TF * IDF$

Token	Nt	IDF	D1		D2		D3		D4		D5		D6		D7	
			TF	**TF-IDF**	TF	**TF-IDF**	TF	**TF-IDF**	TF	**TF-IDF**	TF	**TF-IDF**	TF	**TF-IDF**	TF	**TF-IDF**
3	1	0.8450 9804	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
rugi	1	0.8450 9804	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**
menang	1	0.8450 9804	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
gembira	1	0.8450 9804	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
lebaran	2	0.5440 680444	1	**0.8450 9804**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**
tanggal	1	0.8450 9804	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**
6	1	0.8450 9804	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**
kembang	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**
teknologi	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**
2024	1	0.8450 9804	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**
nasi	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**
sambut	1	0.8450 9804	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
april	1	0.8450 9804	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**
saya	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**
gol	1	0.8450 9804	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
cerdas	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**
penuh	1	0.8450 9804	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
triliun	1	0.8450 9804	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**
lolos	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**
goreng	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**
balas	1	0.8450 9804	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
dunia	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**
libur	1	0.8450 9804	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**
puasa	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**
piala	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**
indonesia	3	0.3679 767853	0	**0**	1	**0.8450 9804**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**

korupsi	1	0.8450 9804	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**
2026	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**
lawan	1	0.8450 9804	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
cepat	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**
harap	1	0.8450 9804	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
vietnam	1	0.8450 9804	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**
makan	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**
peluang	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**
271	1	0.8450 9804	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**
buka	1	0.8450 9804	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**
negara	1	0.8450 9804	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**
timah	1	0.8450 9804	0	**0**	0	**0**	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**
masyarakat	1	0.8450 9804	1	**0.8450 9804**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**

Hasil vektorisasi unigram setiap dokumen diatas adalah sesuai dengan kotak yang dihighlight dengan warna merah seperti:

Untuk dokumen 1 vektorisasi unigram adalah:
$[0\ 0\ 0\ 0.84509804\ 0.84509804\ 0\ 0\ 0\ 0\ 0\ 0\ 0.84509804\ 0\ 0\ 0\ 0\ 0.84509804\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.84509804\ 0\ 0\ 0\ 0\ 0\ 0\ 0.84509804]$
Dan seterusnya…

3. Lakukan pengelompokkan kalimat menggunakan algoritma Hierarchical Agglomerative Clustering dengan menggunakan teknik pengelompokkan smallest distance dan rumus jarak euclidean distance dan cosine similarity (lakukan 2 kali pengelompokkan, 1 menggunakan euclidean distance dan 1 menggunakan cosine similarity) secara manual dan jelaskan setiap langkah yang Anda lakukan dengan lengkap.

   Perhitungan dievaluasi **berdasarkan nilai TF-IDF** yang ditemukan sebelumnya.

   - Euclidian (menggunakan Single Link: closest distance)

   Saya menggunakan rumus euclidian $d(D^a, D^b) = \sqrt{\sum_1^N D^a{}_i - D^b{}_i}$. dimana D^adan D^b adalah dokumen seperti Dokumen 1 (D1), Dokumen 2 (D2), dll.

   Setelah dievaluasi iterasi pertama didapatkan:

Documents	D1	D2	D3	D4	D5	D6	D7
D1	0						
D2	3.047044316	0					
D3	3.047044316	2.927505485	0				
D4	2.672434653	3.047044316	3.047044316	0			
D5	2.672434653	2.80287311	2.80287311	2.672434653	0		
D6	2.927505485	3.047044316	3.047044316	2.927505485	2.672434653	0	
D7	2.927505485	2.80287311	2.80287311	2.927505485	**2.672434653**	2.927505485	0

Selanjutnya karena menggunakan single linkage, kita dapatkan poin dengan jarak terkecil. Ada beberapa poin, namun disini saya pilih D5 dan D7. Oleh karena D7 dan D5 dapat digabungkan. Disini, kita hanya perlu mengisi sesuai **perbandingan D5 dan D7** beserta poin lainnya.

Sebagai contoh **D5 dengan D1** dan **D7 dengan D1** dapat dilihat bahwa jarak **D5 ke D1 lebih kecil** daripada **D7 ke D1**. Oleh karena itu kita ambil jarak terkecilnya yaitu nilai **2.672434653** yang akan kita petakan untuk poin D5, D7 ke D1. Untuk berikutnya saya tidak akan terlalu menjelaskan mengapa pemilihan ini terjadi karena saya rasa sudah cukup jelas.

Berikut adalah iterasi pertama dari algoritma Hierarchical Agglomerative Clustering:

Documents	D1	D2	D3	D4	D5, D7	D6
D1	0					

D2	3.047044316	0				
D3	3.047044316	2.927505485	0			
D4	2.672434653	3.047044316	3.047044316	0		
D5, D7	2.672434653	2.80287311	2.80287311	2.672434653	0	
D6	2.927505485	3.047044316	3.047044316	2.927505485	**2.672434653**	0

Dapat dilihat bahwa D5, D7 dan D6 dapat digabung menjadi ((D5, D7), (D6)).

Sehingga iterasi kedua menghasilkan hasil:

Documents	D1	D2	D3	D4	((D5, D7), (D6))
D1	0				
D2	3.047044316	0			
D3	3.047044316	2.927505485	0		
D4	2.672434653	3.047044316	3.047044316	0	
((D5, D7), (D6))	2.672434653	2.80287311	2.80287311	**2.672434653**	0

Dapat dilihat bahwa ((D5, D7), (D6)) dan D4 dapat digabung menjadi (((D5, D7), (D6)), (D4))

Sehingga iterasi ketiga menghasilkan hasil:

Documents	D1	D2	D3	((D5, D7), (D6))(D4))
D1	0			
D2	3.047044316	0		
D3	3.047044316	2.927505485	0	
((D5, D7), (D6))(D4))	**2.672434653**	2.80287311	2.80287311	0

Dapat dilihat bahwa (((D5, D7), (D6)), (D4)) dan (D1) dapat digabung menjadi ((((D5, D7), (D6)), (D4)), D1)

Sehingga iterasi keempat menghasilkan hasil:

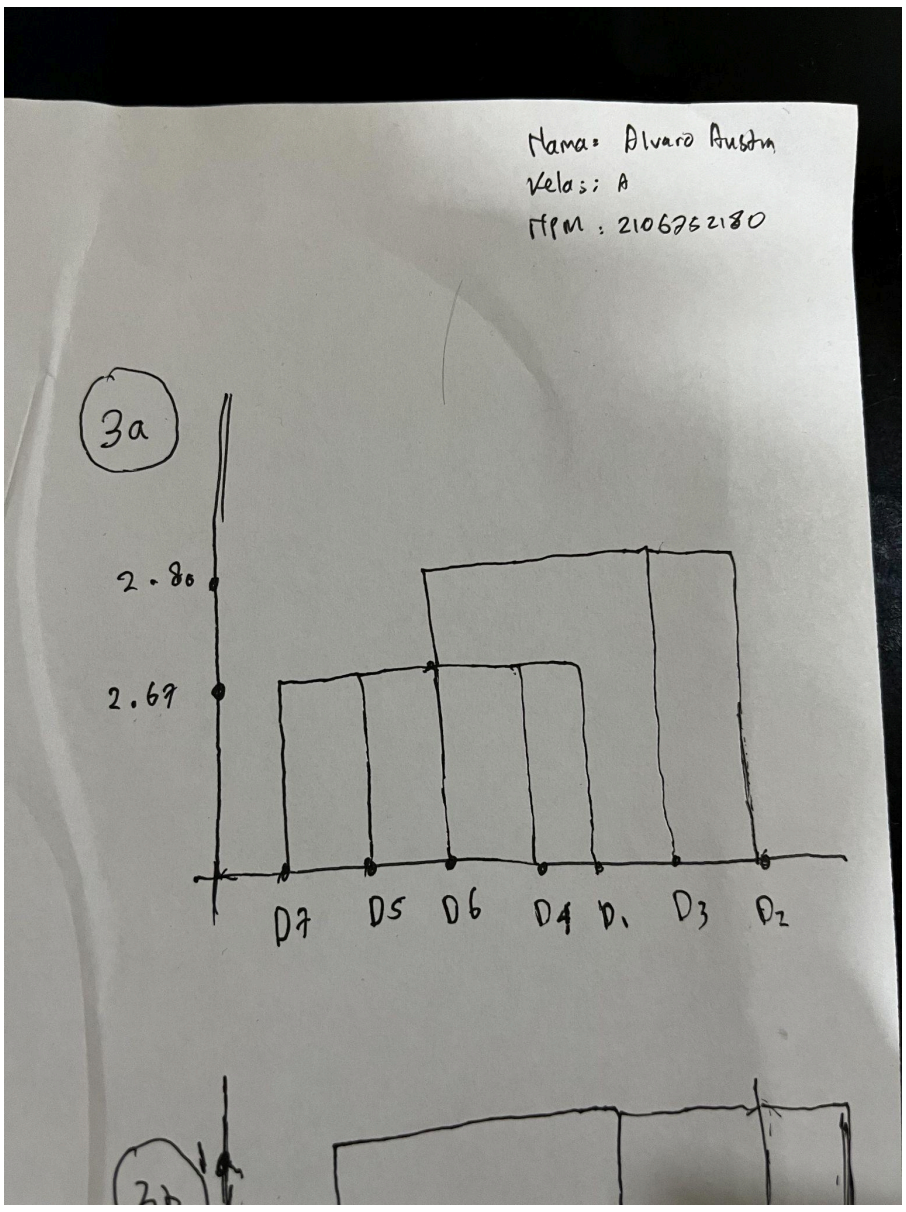
Documents	((((D5, D7), (D6)), (D4)), D1)	D2	D3
((((D5, D7), (D6)), (D4)), D1)	0		
D2	2.80287311	0	
D3	**2.80287311**	2.927505485	0

Dapat dilihat bahwa ((((D5, D7), (D6)), (D4)), D1) dan (D3) dapat digabung menjadi (((((D5, D7), (D6)), (D4)), D1), D3)

Documents	(((((D5, D7), (D6)), (D4)), D1), D3)	D2	D3
(((((D5, D7), (D6)), (D4)), D1), D3)	0		
D2	**2.80287311**	0	

Akhirnya sudah sampai tahap terakhir dimana kita berada pada iterasi terakhir dan semuanya dapat digabung.
Hasil akhir menjadi ((((((D5, D7), (D6)), (D4)), D1), D3), (D2))

Dendogram untuk algoritma menggunakan euclidian dapat dilihat disini:



- Cosine Similarity
  Untuk melakukan perhitungan cosine distance, akan dilakukan operasi seperti berikut:
  $Cos(A, B) = \frac{A.B}{||A|| \, ||B||}$ dimana hasil dari Cos(A, B) dikurangi 1, maka
  Cosine distance = 1 - Cosine Similarity

  Dengan menggunakan rumus berikut, kita akan menggunakan vektor A dan B berupa TF-IDF dari D1, D2, …, D7.
  Berikut adalah hasil awal-awal yang pemetaan D1 dengan D1, D2, dengan D1, D2 dengan D2, dst…

Documents	D1	D2	D3	D4	D5	D6	D7
D1	0						
D2	1	0					
D3	1	0.8571428571	0				
D4	**0.8333333333**	1	1	0			
D5	1	1	1	1	0		
D6	1	1	1	1	1	0	
D7	1	0.84569665	0.84569665	1	1	1	0

Pada soal ini, saya akan menggunakan distance terkecil untuk melakukan algoritma clustering ini. Dapat dilihat bahwa D4 dan D1 memiliki distance terkecil, oleh karena itu kita akan menggabungkan kedua dokumen tersebut menjadi (D1, D4).

Selanjutnya untuk iterasi pertama, kita akan memperoleh:

Documents	(D1, D4)	D2	D3	D5	D6	D7
(D1, D4)	0					
D2	1	0				
D3	1	0.8571428571	0			
D5	1	1	1	0		
D6	1	1	1	1	0	
D7	1	0.84569665	**0.84569665**	1	1	0

Dimana dapat dilihat bahwa D7 dan D3 dapat digabungkan menghasilkan (D3, D7). Lakukan lagi pemindaian untuk melihat dokumen mana yang paling kecil dengan D3 dan D7, lalu dipilih yang terkecil.

Hasil iterasi kedua, kita akan memperoleh:

Documents	(D1, D4)	D2	(D3, D7)	D5	D6
(D1, D4)	0				
D2	1	0			
(D3, D7)	1	**0.84569665**	0		
D5	1	1	1	0	
D6	1	1	1	1	0

Dapat dilihat bahwa (D3, D7) dan D2 memiliki jarak terkecil sehingga dapat kita gabungkan menghasilkan (D3, D7), (D2)

Hasil iterasi ketiga, kita akan memperoleh:

Documents	(D1, D4)	(D3, D7), (D2)	D5	D6
(D1, D4)	0			
(D3, D7), (D2)	**1**	0		
D5	1	1	0	
D6	1	1	1	0

Dapat dilihat bahwa kita dapat menggabungkan (D1, D4) dengan (D3, D7), (D2) karena memiliki jarak terkecil yaitu 1 menjadi (D1, D4) ((D3, D7), D2).

Berikut adalah hasil iterasi keempat:

Documents	(D1, D4) ((D3, D7), D2)	D5	D6
(D1, D4) ((D3, D7), D2)	0		
D5	**1**	0	
D6	1	1	0

Selanjutnya dapat dilihat bahwa (D1, D4) ((D3, D7), D2) bisa digabungkan dengan D5 karena jaraknya terkecil. Oleh karena itu akan menghasilkan ((D1, D4) ((D3, D7), D2)), (D5).

Terakhir ini adalah hasil dari iterasi terakhir:

Documents	((D1, D4) ((D3, D7), D2)), (D5)	D6
((D1, D4) ((D3, D7), D2)), (D5)	0	
D6	**1**	0

Sehingga akhirnya kita dapat menggabungkan seluruh nya dan algoritma clustering berhasil menjadi (((D1, D4) ((D3, D7), D2)), (D5)), (D6).

Berikut adalah gambar dendogram untuk bagian cosine similarity ini: