BAB II LANDASAN TEORI

Pada bab ini akan dijelaskan landasan teori yang digunakan pada pengerjaan tugas akhir ini. Landasan teori meliputi penjelasan mengenai pengelompokan dokumen, teknik-teknik yang digunakan serta penelitian yang telah dilakukan terkait dengan pengelompokan dokumen.

2.1. Pengelompokan Dokumen atau *Document Clustering*

Pengelompokan dokumen adalah mengelompokkan dokumen-dokumen dalam koleksi ke dalam kategori atau kluster sesuai dengan kemiripan isi antar dokumen. Pengelompokan dokumen dapat dijadikan cara untuk organisasi dan perolehan informasi (*retrieval*). Dengan teknik pengelompokan dokumen yang baik, komputer dapat secara otomatis mengelompokkan dokumen ke dalam kluster yang bisa mendukung proses pencarian yang lebih efisien. Hal ini akan sangat berguna ketika jumlah dokumen dalam koleksi sangat besar. Dokumen akan lebih mudah ditemukan jika telah dikelompokkan ke dalam kluster-kluster.

Dalam *document clustering,* dokumen-dokumen yang mirip akan dikelompokkan ke dalam kluster yang sama. Pengelompokan ini dapat dilakukan dengan mengidentifikasi kata-kata khusus yang muncul di dalam dokumen. Kumpulan kata-kata khusus ini dapat digunakan untuk mengukur tingkat kemiripan isi antar dokumen satu dengan dokumen lainnya. Dengan demikian dokumen-dokumen dengan tingkat kemiripan yang tinggi akan dikelompokkan ke dalam satu kluster. Ilustrasi dari penjelasan diatas dapat diberikan dalam contoh berupa kalimat dalam dokumen berikut ini:

1. Kalangan investor di bursa saham Wall Street, New York, menantikan langkah baru yang akan diambil untuk menyelamatkan industri keuangan.

2. Investor bursa saham regional pada awal perdagangan Selasa (10/2) pagi bergerak hati-hati menanti perkembangan paket stimulus ekonomi Amerika Serikat yang masih ada di tangan Kongres.

3. Setelah gagal di dua laga sebelumnya, Bima Sakti Nikko Steel Malang, tuan rumah seri dua putaran pertama Indonesian Basketball League 2009, Senin (9/2) berhasil memetik kemenangan kandang pertama dengan skor 63-61.

4. Di laga lanjutan seri dua putaran pertama IBL 2009 di GOR Bima Sakti Malang, Jawa Timur, Minggu (8/2), SM Britama memetik kemenangan atas rival utamanya XL ASPAC Jakarta dengan skor 74-63.

Pada kalimat pertama terdapat kata-kata khusus seperti investor, bursa, saham, dan keuangan yang sebagian besar muncul juga pada kalimat kedua tetapi tidak muncul pada kalimat ketiga dan keempat. Sedangkan pada kalimat ketiga terdapat kata-kata khusus seperti laga, seri, skor, kemenangan yang semuanya muncul juga pada kalimat keempat. Dengan demikian, dapat disimpulkan bahwa kalimat pertama lebih mirip dengan kalimat kedua dibandingkan dengan kalimat ketiga dan keempat. Begitu juga dapat disimpulkan bahwa kalimat ketiga lebih mirip dibandingkan dengan kalimat keempat. Jadi, kalimat pertama dan kalimat kedua dapat dikelompokkan ke dalam satu kluster dan kluster lainnya mengandung kalimat ketiga dan keempat.

Kata-kata khusus yang muncul secara bersamaan dapat dijadikan informasi dalam menentukan kluster suatu dokumen. Oleh karena itu, dalam pemodelan *semantic* dari dokumen, keterkaitan kata-kata ini digunakan dalam merepresentasikan suatu dokumen yang kemudian digunakan untuk menentukan tingkat kemiripan dokumen satu dengan dokumen lain.

Pengelompokan dokumen pada dasarnya dapat dilakukan dengan dua cara yaitu klasifikasi dan *clustering* (pengelompokan). Klasifikasi adalah cara mengelompokkan dokumen yang menggunakan pendekatan *supervised* (Dunham, 2003). Dengan pendekatan ini, informasi mengenai kategori atau topik untuk klasifikasi sudah didefinisikan terlebih dahulu.

Sedangkan *clustering* adalah cara pengelompokan yang menggunakan pendekatan *unsupervised*. Dengan pendekatan ini, tidak ada informasi mengenai data yang dipakai yaitu mengenai kategori atau topik yang digunakan. Pengelompokan dokumen akan langsung dilakukan dengan melihat pola yang ada pada koleksi

sehingga pengelompokkan dokumen ke dalam kluster dilakukan berdasarkan tingkat kemiripan (Dunham, 2003).

2.2. Teknik untuk Mengelompokkan Dokumen

Dalam proses pengelompokan dokumen, ada dua pendekatan yang digunakan yaitu *supervised* dan *unsupervised*. Jika pada proses pengelompokan, proses pembelajaran dilibatkan sebelum proses pengelompokan itu sendiri dilakukan, maka pendekatan ini adalah pendekatan *supervised*. Sebelum melakukan pengelompokan dokumen, diperlukan pembelajaran mengenai ciri-ciri dokumen yang termasuk dalam suatu topik. Pembelajaran ini dilakukan pada fase *training* dimana model akan dibangun yang nantinya akan digunakan untuk mengklasifikasikan dokumen pada fase *testing*. Pada awalnya dilakukan pemilihan fitur yang akan digunakan pada pembangunan model dan pengklasifikasian. Salah satu fitur yang digunakan adalah fitur yang menggunakan konsep *n-gram*. Pada konsep ini fitur yang digunakan berupa n buah kata yang muncul berurutan pada suatu dokumen. Fitur ini yang menjadi dasar dari model yang dibangun. Telah banyak teknik yang telah dikembangkan dengan pendekatan *supervised* untuk mengklasifikasikan dokumen seperti Naïve Bayes.

Pendekatan yang lain adalah pendekatan *unsupervised*. Dalam pendekatan ini tidak ada proses pembelajaran. Pengelompokan dokumen dilakukan langsung pada koleksi dokumen dengan mempelajari pola yang ada pada koleksi dokumen. (Berry & Shahnaz, 2004). Telah banyak teknik pengelompokan dokumen dengan pendekatan ini yang telah dikembangkan. Salah satunya yang cukup banyak digunakan adalah K-Means. Teknik ini mengelompokkan dokumen dengan cara memilih pusat kluster dan secara iteratif memindahkan dokumen ke dalam kluster yang paling dekat dengannya hingga tidak ada lagi dokumen yang dipindahkan. Namun, ada tantangan yang cukup besar yang dihadapi yaitu masalah dimensi dari data yang digunakan karena banyak teknik yang menggunakan representasi matriks dalam penerapannya dan biasanya biaya komputasi akan sangat besar jika jumlah data sangat besar yang membuat dimensi menjadi sangat tinggi.

Salah satu solusi yang bisa dilakukan adalah mengurangi dimensi dari data tetapi tetap menjaga integritas informasi yang ada didalamnya. Selain mengatasi

masalah komputasi, reduksi dimensi juga dapat menghasilkan representasi *semantic* yang lebih akurat seperti yang dilakukan oleh teknik Latent Semantic Analysis (LSA). Salah satu teknik yang bisa digunakan untuk reduksi dimensi adalah Random Projection. Teknik ini akan mereduksi dimensi dari data yang digunakan tetapi tetap menjaga informasi yang terkandung didalamnya (Lin & Gunopulos, 2003). Hasil dari reduksi dimensi ini baru kemudian digunakan oleh teknik pengelompokan dokumen seperti K-Means.

Struktur pengelompokan dokumen yang digunakan bisa berupa partisi (*partitional*) ataupun hierarki (*hierarchical*) (Dunham, 2003). Pada struktur partisi, dokumen dalam koleksi akan dikelompokkan ke dalam kluster sejajar yang saling *disjoint* sesuai dengan pola yang ada dalam koleksi. Sedangkan pada struktur hierarki, organisasi dari dokumen yang ada berupa struktur *tree* dimana posisi *root* ditempati oleh koleksi dokumen. Pada level-level dibawahnya, koleksi tersebut akan dipecah menjadi grup yang lebih kecil sesuai dengan kriteria pembagian yang ditentukan. Contoh teknik pengelompokan dokumen yang menggunakan struktur partisi adalah K-Means dan Non-Negative Matrix Factorization.

Teknik Non-Negative Matrix Factorization adalah teknik yang bisa digunakan untuk mengelompokkan dokumen dan telah mengatasi masalah dimensi. Teknik ini menggunakan struktur partisi dan mengelompokkan dokumen sesuai dengan *semantic feature* yang ada didalamnya. Pada percobaan yang dilakukan oleh Berry & Shahnaz (2004), teknik Non-Negative Matrix Factorization digunakan untuk mengelompokkan koleksi dokumen ke dalam kluster sesuai dengan pola yang ada dalam koleksi. Pada percobaan tersebut, teknik ini bisa mengelompokkan dokumen dengan akurasi hingga mencapai diatas 90 persen. Pada percobaan Xu, Liu, & Gong (2001) teknik Non-Negative Matrix Factorization, dengan menggunakan informasi fitur berupa TF-IDF (*term frequency-inverse document frequency*), bisa mengelompokkan dokumen dengan akurasi mencapai diatas 80 persen.

2.3. *Vector Space Model*

Sebelum membahas mengenai teknik pengelompokan dokumen, perlu diketahui bagaimana data direpresentasikan. Semua teknik yang digunakan untuk

pengelompokan dokumen dalam percobaan ini menggunakan representasi model yang sama yaitu *vector space model*.

Pada *vector space model,* setiap dokumen direpresentasikan dengan vektor dalam ruang vektor (Berry, Drmac, & Jessup, 1999). Lebih lanjut dijelaskan pada (Berry & Shahnaz, 2004) vektor dokumen ini adalah vektor berdimensi m dimana m adalah jumlah *term* yang digunakan dalam model tersebut. Setiap komponen dalam vektor merefleksikan tingkat dimana *term* yang bersangkutan memberikan arti (*semantic*) pada dokumen tersebut. Dalam penerapannya, nilai dari masing-masing komponen vektor adalah frekuensi kemunculan *term* yang bersangkutan pada dokumen. Dengan pengertian ini, maka koleksi dokumen dapat dinyatakan dengan *term-document matrix* yang merupakan penggabungan dari masing-masing vektor dokumen. *Term-document matrix* m x n ini terdiri dari n buah vektor dokumen yang masing-masingnya berdimensi m dimana m adalah jumlah *term.* Masing-masing komponen dalam matriks ini adalah frekuensi kemunculan *term* yang bersangkutan pada dokumen atau dengan kata lain:

Untuk m x n *term-document matrix X* : *X(i,j)* = frekuensi kemunculan *term* ke-*i* pada dokumen ke-*j*

Arah dari vektor bisa digunakan untuk mengindikasikan tingkat kesamaan arti atau *semantic* antar dokumen yang direpresentasikan oleh vektor yang bersangkutan (Sahlgren, 2005). Dua dokumen yang memiliki *term* yang sama yang muncul didalamnya akan memberikan pengertian bahwa dokumen tersebut berhubungan dengan hal yang sama. Contohnya bandingkan tiga kalimat berikut ini: (Contoh 2.1)

1. Kalangan investor di bursa saham Wall Street, New York, menantikan langkah baru yang akan diambil untuk menyelamatkan industri keuangan.

2. Investor saham di Indonesia perlu inisiatif baru yang perlu diambil untuk menyelamatkan industri keuangan negara.

3. Valentino Rossi mengukir hasil yang memuaskan saat melakukan tes di Sirkuit Sepang, Malaysia.

Jika dibandingkan dari segi kemunculan *term*, terlihat bahwa kalimat 1 dan kalimat 2 membicarakan hal yang sama yaitu mengenai bisnis keuangan. Sedangkan terlihat bahwa kalimat 1 dan kalimat 3 membicarakan hal yang berbeda. Hal ini jika digambarkan dalam ruang vektor, maka kalimat 1 dan kalimat 2 akan memiliki arah yang hampir sama dibandingkan dengan arah antara vektor kalimat 1 dan kalimat 3.

Representasi *term-document matrix* dari 3 kalimat pada Contoh 2.1 diatas dengan fitur berupa *term* investor, bursa, saham, industri, keuangan, Valentino, Rossi, tes, dan sirkuit adalah sebagai berikut: (D1 = kalimat 1; D2 = kalimat 2; D3 = kalimat 3)

	D1	D2	D3
investor	1	1	0
Bursa	1	0	0
Saham	1	1	0
Industri	1	1	0
Keuangan	1	1	0
valentino	0	0	1
rossi	0	0	1
tes	0	0	1
Sirkuit	0	0	1

Gambar 2.1. Contoh *Term-document Matrix*

Ada beberapa alasan mengapa *vector space model* cukup banyak digunakan dalam merepresentasikan data. Alasan tersebut antara lain: (Sahlgren, 2005)

1. *Vector space* terdefinisi dengan baik secara matematis dan mudah untuk dipahami. Perilaku dari *vector space* sendiri telah diketahui. Selain itu telah banyak *tools* yang bisa digunakan untuk memanipulasinya.

2. Dengan model ini, kesamaan atau *similarity* arti bisa dengan mudah ditentukan secara matematis.

Walaupun model ini cukup banyak digunakan, model ini memiliki masalah dalam hal efisiensi dan skalabilitas. *Term-document matrix* yang ukurannya ditentukan oleh jumlah dokumen dan jumlah *term* akan memiliki ukuran yang sangat besar ketika jumlah dokumen dalam koleksi sangat besar yang membuat jumlah *term* juga menjadi besar. Selain itu, matriks ini akan memiliki *cell* yang sebagian besar

akan bernilai nol yang membuat matriks ini menjadi matriks yang *sparse*. Hal ini dikarenakan untuk setiap dokumen, hanya akan sedikit dari koleksi *term* yang muncul dalam dokumen tersebut. Bahkan disebutkan pada (Sahlgren) bahwa menurut *Zipf's Law,* 99% dari *cell* matriks ini akan bernilai nol. Oleh karena itu, untuk mengatasi masalah dimensi tinggi dan *sparseness* ini, diperlukan teknik reduksi dimensi. Pada subbab selanjutnya akan dijelaskan mengenai teknik pengelompokan dokumen yang menggunakan teknik reduksi dimensi.

2.4. Metrik untuk Pengukuran Tingkat Kesamaan

Dengan representasi data dalam *vector space model* dimana setiap dokumen direpresentasikan sebagai vektor, maka tingkat kesamaan atau *similarity* antar dokumen pun bisa dihitung dengan mudah. Pengukuran tingkat kesamaan yang umum digunakan adalah *Euclidean distance* dan *Cosine Angle* antar vektor.

Euclidean Distance antara dua dokumen *x* dan *y* yang direpresentasikan sebagai vektor *x* dan vektor *y* dapat didefinisikan sebagai:

$$\|x - y\| = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{1}$$

Semakin besar nilai *Euclidean Distance* antara vektor *x* dan vektor *y*, maka tingkat kesamaan antara dokumen *x* dan *y* yang direpresentasikan oleh vektor tersebut semakin kecil. Untuk Contoh 2.1 pada subbab 2.3, *Euclidean Distance* antar masing-masing vektor dokumen adalah sebagai berikut: (D1 = kalimat 1; D2 = kalimat 2; D3 = kalimat 3)

Euclidean_distance(D1,D2) = 1
Euclidean_distance(D1,D3) = 3
Euclidean_distance(D2,D3) = 2.8284

Dari informasi diatas, dapat disimpulkan bahwa D1 memiliki tingkat kesamaan yang lebih tinggi dengan D2 daripada dengan D3 karena nilai *Euclidean Distance* antara D1 dan D2 (yaitu 1) lebih kecil dibandingkan dengan nilai *Euclidean Distance* antara D1 dan D3 (yaitu 3). Penjelasan ini juga berlaku pada D2 dan D3.

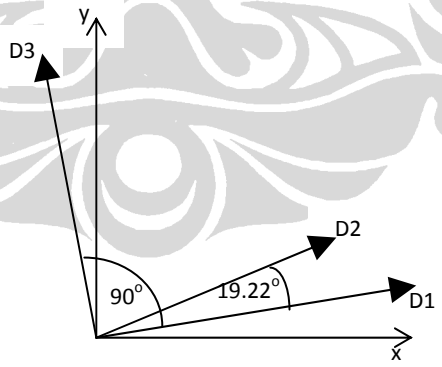
Sedangkan *Cosine Angle* antar dua dokumen *x* dan *y* yang direpresentasikan sebagai vektor *x* dan vektor *y* dapat didefinisikan sebagai: (Lin & Gunopulos, 2003)

$$sim(x, y) = \cos(\theta) = \sum_{i=0}^{m} \frac{x_i . y_i}{\|x\| \cdot \|y\|} = \frac{x \cdot y}{\|x\| \cdot \|y\|} \quad (2)$$

Semakin kecil nilai *angle* (Θ) antar vektor *x* dan *y*, maka tingkat kesamaan antara dokumen *x* dan *y* yang direpresentasikan oleh vektor tersebut semakin besar. Untuk Contoh 2.1 pada subbab 2.3, *Cosine Angle* antar masing-masing vektor dokumen adalah sebagai berikut: (D1 = kalimat 1; D2 = kalimat 2; D3 = kalimat 3)

Cosine_angle(D1,D2) = 0.9443
Cosine_angle(D1,D3) = 0
Cosine_angle(D2,D3) = 0

Dari informasi diatas, dapat diketahui bahwa *angle* antara vektor dokumen D1 dengan D2 (19.22^o) lebih kecil daripada *angle* antara vektor dokumen D1 dengan D3 (90^o) sehingga dapat disimpulkan bahwa D1 dan D2 memiliki tingkat kesamaan yang lebih tinggi dibandingkan D1 dan D3. Ilustrasi dari penjelasan diatas ditunjukkan pada Gambar 2.2.



Gambar 2.2. Ilustrasi Vektor Dokumen dalam Sistem Koordinat

Dari definisi dan sifatnya, kedua jenis pengukuran diatas dapat digunakan dalam pengelompokan dokumen dalam menentukan tingkat kesamaan antar dokumen. Hal ini menjadi dasar dalam penentuan kluster dari masing-masing dokumen dalam koleksi.

2.5. Nonnegative Matrix Factorization

Teknik Non-Negative Matrix Factorization dapat didefinisikan sebagai berikut: (Lee & Seung, 2001)

Diberikan non-negatif matriks V, cari non-negatif matriks W dan H sedemikian sehingga $V \approx W H$ (3)

Sebagai ilustrasinya adalah diberikan himpunan vektor berdimensi m dan vektor dikumpulkan dalam satu matriks V berukuran m x n dimana n adalah jumlah vektor. Matriks ini akan difaktorisasi menjadi matriks W berukuran m x r dan matriks H berukuran r x n. Biasanya r adalah nilai yang jauh lebih kecil dari n dan m (r << m & n) sehingga matriks W dan H akan lebih kecil dari pada matriks awal V.

Persamaan (3) dapat ditulis dalam bentuk kolom per kolom sebagai $v = W h$ dimana v dan h adalah kolom-kolom dari matriks V dan H. Dengan kata lain, setiap vektor v diaproksimasi dengan kombinasi linier kolom W dengan bobot komponen h. Dengan demikian, W dapat dianggap sebagai matriks yang mengandung basis yang dapat digunakan untuk mengaproksimasi data yang ada di matriks V. Sedangkan H mengandung informasi kombinasi linier dari basis untuk mengaproksimasi matriks awal V. Dari penjelasan diatas dapat disimpulkan bahwa Non-Negative Matrix Factorization merupakan teknik reduksi dimensi dimana untuk merepresentasikan data yang lebih banyak yaitu sebanyak m, hanya digunakan basis yang sedikit, yaitu r dimana r jauh lebih kecil dari m.

Pada penerapannya sebagai teknik pengelompokan yang bersifat *unsupervised* dan *partitional*, Non-Negative Matrix Factorization memproses koleksi dokumen sebagai *term-document matrix*. Sifat non-negatif yang dibawa oleh *term-document matrix* akan tetap dipertahankan selama proses faktorisasi yang menghasilkan faktor yang lebih kecil dimensinya. Matriks W akan mengandung basis berupa basis dari tiap kluster dokumen. Kemudian kluster dari suatu dokumen dapat ditentukan dengan cara mencari basis dimana dokumen tersebut memiliki nilai proyeksi yang paling besar. Dengan kata lain, untuk menentukan kluster suatu dokumen, yang perlu dilakukan adalah mencari nilai maksimum pada matriks H untuk dokumen yang bersangkutan. Dengan demikian, untuk pengelompokan

dokumen, Non-Negative Matrix Factorization tidak memerlukan proses tambahan. Kluster dari suatu dokumen bisa ditentukan langsung dari hasil faktorisasi.

Matrik *W* dan *H* dibentuk dengan meminimalkan *Euclidean norm* dari *V-WH*. Problem Non-Negative Matrix Factorization kemudian dapat dinyatakan sebagai berikut:

Misalkan $V \in R^{nxm}$ adalah matriks non-negatif dan $W \in R^{mxk}$ dan $H \in R^{kxn}$ untuk $0 < k << min(m,n)$. Kemudian, *objective function* dari problem ini dapat dinyatakan sebagai:

$$\min_{W,H} \|V - WH\|_F^2, W_{ij} > 0, H_{ij} > 0 \tag{4}$$

Matrik *H* dan *W* dibentuk dengan cara iteratif sampai konvergen dengan inisialisasi awal berupa matriks acak. Salah satu *update rule* untuk matriks *W* dan *H* yang diusulkan oleh Lee & Seung (2001) disebut dengan *Multiplicative Method* (MM). *Objective function* yang ingin dicapai adalah

$$\min J = \|V - WH\|^2 \tag{5}$$

$$\begin{aligned} J &= (V - WH)(V - WH)^T \\ J &= \left(VV^T - 2VH^TW^T + WHH^TW^T\right) \end{aligned} \tag{6}$$

Misalkan $W = [w_{ij}]$, $H = [h_{xy}]$, atau *W* sebagai kumpulan vektor kolom $W = [W_1, W_2, ..., W_k]$, problem diatas dapat dinyatakan sebagai meminimalkan *J* dalam hubungannya dengan *W* dan *H* dengan *constraint* $w_{ij} \geq 0$, $h_{xy} \geq 0$, dimana $0 \leq i \leq m$, $0 \leq j \leq k$, $0 \leq x \leq k$, $0 \leq j \leq n$. Problem ini adalah problem optimisasi berkendala (*constrained optimization problem*) dan dapat diselesaikan dengan *Lagrange multiplier method*. Misalkan α_{ij} dan β_{ij} adalah berturut turut *Lagrange multiplier* untuk *constraint* $w_{ij} \geq 0$ dan $h_{xy} \geq 0$ dan $\alpha = [\alpha_{ij}]$, $\beta = [\beta_{ij}]$, *Lagrange L* adalah

$$L = J + \alpha W + \beta H \tag{7}$$

Turunan pertama fungsi *L* terhadap *W* dan *H* masing-masing adalah

$$\frac{\partial L}{\partial W} = -VH^T + WHH^T + \alpha \tag{8}$$

$$\frac{\partial L}{\partial H} = -W^T V + W^T WH + \beta \tag{9}$$

Dengan menggunakan *Kuhn-Tucker condition* $\alpha_{ij}w_{ij}=0$ dan $\beta_{ij}h_{ij}=0$, maka didapat persamaan sebagai berikut:

$$(VH^T)_{ij} w_{ij} - (WHH^T)_{ij} w_{ij} = 0 \tag{10}$$

$$(W^T V)_{ij} h_{ij} - (W^T WH)_{ij} h_{ij} = 0 \tag{11}$$

Dari persamaan (10) dan (11) dapat dibuat *update rule* sebagai berikut:

$$w_{ij} = w_{ij} \frac{(VH^T)_{ij}}{(WHH^T)_{ij}} \tag{12}$$

$$h_{ij} = h_{ij} \frac{(W^T V)_{ij}}{(W^T WH)_{ij}} \tag{13}$$

Pada implementasinya pembilang dari masing-masing *update rule* ditambah dengan suatu konstanta ε misalkan 10^{-9} untuk menghindari kasus pembagian dengan nol. Kompleksitas dari *update rule* ini adalah *O(kmn)* untuk setiap iterasi dengan n adalah jumlah *term,* m adalah jumlah dokumen dan k adalah jumlah kluster yang ingin dibentuk.

Pada (Lee & Seung, 2001) telah dibuktikan bahwa nilai *objective function J* tidak naik (*non-increasing*) dengan *update rule* diatas. Selain itu, dengan *update rule* tersebut, konvergensi juga dijamin yaitu kondisi dimana matriks W dan H pada iterasi ke *k* akan memiliki perbedaan yang sangat kecil dan dapat dianggap sama dengan matriks *W* dan *H* pada iterasi ke *(k-1)* sehingga iterasi bisa dihentikan (konvergen).

Setiap elemen w_{ij} pada matriks W merepresentasikan tingkatan dimana *term* yang bersangkutan (*term* ke-*i*) termasuk dalam kluster *j*. Sedangkan setiap elemen h_{ij} pada matriks *H* mengindikasikan tingkatan dimana dokumen ke-*j* berasosiasi dengan

kluster *i*. Jika dokumen ke-*j* termasuk dalam kluster *x*, maka h_{xi} akan memiliki nilai yang besar sedangkan elemen lain pada vektor ke-*i* pada matriks H akan memiliki nilai yang sangat kecil dan mendekati nol. Dari pengertian ini, pengelompokan dokumen pun dapat dilakukan berdasarkan informasi yang ada pada matriks H.

Selain *Multiplicative Method* (MM), terdapat metode *hybrid* yang menggabungkan MM, yang merupakan *gradient descent optimization problem,* dengan *constrained least square* (CLS) *model.* Metode yang dinamakan dengan GD-CLS ini diusulkan oleh Berry & Shahnaz (2004). Pada metode ini, matriks *W* di-*update* dengan menggunakan metode MM sedangkan matriks *H* di-*update* menggunakan model CLS. Untuk setiap iterasi, matriks *H* didapatkan dengan menyelesaikan *constrained least square problem*:

$$\min J = \min \left\{ \left\| V_j - WH_j \right\|_2^2 + \lambda \left\| H_j \right\|_2^2 \right\}, j = 1,...., n \tag{14}$$

$$\begin{aligned} J &= \left(V_j - WH_j\right)^T \left(V_j - WH_j\right) + \lambda H_j^T H_j \\ J &= H_j^T W^T W H_j - 2H_j^T W^T V_j + V_j^T V_j + \lambda H_j^T H_j \end{aligned} \tag{15}$$

Selanjutnya nilai H_j yang meminimumkan *J* adalah penyelesaian dari

$$\begin{aligned} &\frac{\partial J}{\partial H_j} = 0 \\ &W^T W H_j - W^T V_j + \lambda H_j = 0 \\ &H_j = \frac{W^T V_j}{W^T W + \lambda} \end{aligned} \tag{16}$$

Secara ringkas cara kerja pengelompokan dokumen dengan menggunakan teknik reduksi dimensi Nonnegative Matrix Factorization adalah sebagai berikut:

1. Jika diberikan koleksi dokumen dalam korpus, konstruksikan *term-document matrix V* dimana setiap kolom *j* merupakan *term-frequency vector* dari dokumen ke-*j*.

2. Gunakan metode GD-CLS untuk memfaktorkan matriks *V* menjadi matriks *W* dan *H*.

3. Gunakan matriks *H* untuk menentukan kluster masing-masing dokumen dalam korpus dengan cara memeriksa setiap baris *i* untuk setiap *term-frequency vector* dokumen ke-*j*. Dokumen ke-*j* dimasukkan ke dalam kluster *x* jika $x = \arg\max_i h_{ij}$

Untuk Contoh 2.1 pada subbab 2.3 dengan *term-document matrix* yang ditunjukkan pada Gambar 2.1, matriks *W* dan *H* yang dihasilkan dengan metode GD-CLS (dengan λ = 0.01 dan jumlah kluster yang dibentuk k = 2) adalah sebagai berikut:

W =

1.09	0
0.58	0
1.09	0
1.09	0
1.09	0
0	0.99
0	0.99
0	0.99
0	0.99

H =

D1	D2	D3
0.97	0.85	0
0	0	1

Dari matrik *H*, kluster untuk tiap dokumen dapat ditentukan dengan mengikuti langkah ke-3 diatas. Kluster yang didapat adalah dokumen D1 dan D2 termasuk dalam 1 kluster, dan D3 berada pada kluster yang lain.

2.6. Random Projection

Random Projection adalah salah satu teknik reduksi dimensi. Teknik reduksi dimensi seperti SVD (*Singular Value Decomposition*) memerlukan biaya yang cukup besar dalam mereduksi dimensi data. Namun, Random Projection menawarkan reduksi dimensi data dengan biaya yang lebih kecil (Sitbon & Bruza, 2008). Random Projection juga menggunakan input berupa *term-document matrix*. Matriks ini adalah matriks *sparse* yang dihitung dari korpus. Ide awal dari Random Projection adalah memproyeksikan representasi vektor yang *sparse* menjadi lebih *dense* dengan dimensi yang lebih kecil. Ide ini berangkat dari *lemma* yang dikemukan oleh Johnson-Lindenstrauss yaitu himpunan dari n vektor berdimensi tinggi dalam ruang *Euclidean* dapat dipetakan ke dimensi yang lebih rendah dengan basis sembarang

(*random*) dan jarak relatif antar sembarang 2 vektor tetap dipertahankan. (Dasgupta & Gupta, 1999).

Input untuk proses reduksi Random Projection adalah *term-document matrix* M *m x n* dengan *m* adalah jumlah *term* yang ada pada koleksi dokumen dan *n* adalah jumlah dokumen. Matriks ini akan direduksi menjadi matriks M *k x n* dengan *k* adalah dimensi baru dan jauh lebih kecil dari m (k << m) dan *n* adalah jumlah dokumen. Tahapan yang perlu dilalui untuk melakukan reduksi tersebut adalah sebagai berikut: (Sitbon & Bruza, 2008)

1. Buat matriks kosong dimana setiap baris merepresentasikan *term* dalam koleksi dengan kolom berdimensi baru yaitu *k.* Jadi setiap vektor *term* berukuran *1 x k.*
2. Setiap vektor *term* diisi secara acak dengan *k/6* nilai positif dan *k/6* nilai negatif, dan sisanya nol.
3. Buat matriks baru dimana setiap barisnya merepresentasikan dokumen yang ada dengan kolom berdimensi *k.* Setiap vektor dokumen *i* diisi dengan vektor *term j* secara aditif setiap kali dokumen *i* mengandung *term j.*
4. Matriks baru yang dibentuk pada tahap (3) merupakan matriks yang dimensinya telah direduksi.

Ilustrasi dari tahap-tahap reduksi dimensi diatas dapat digambarkan sebagai berikut:

Misalkan terdapat kumpulan 3 dokumen seperti pada Contoh 2.1 (subbab 2.3):

D1: Kalangan investor di bursa saham Wall Street, New York, menantikan langkah baru yang akan diambil untuk menyelamatkan industri keuangan.
D2: Investor saham di Indonesia perlu inisiatif baru yang perlu diambil untuk menyelamatkan industri keuangan negara.
D3: Valentino Rossi mengukir hasil yang memuaskan saat melakukan tes di Sirkuit Sepang, Malaysia.

Misalkan dari *term-term* yang muncul pada keempat dokumen diatas, *term* yang dipakai dalam penghitungan adalah investor, bursa, saham, industri, keuangan, valentino, rossi, tes, dan sirkuti dan dimensi baru ditentukan adalah 6. Maka ada 9 vektor *term* dengan dimensi 6 yang membentuk sebuah matriks acak T sebagai berikut:

T1 (investor) :	0 0 1 0 0 -1
T2 (bursa) :	1 0 0 -1 0 0
T3 (saham) :	0 1 0 -1 0 0
T4 (industri) :	0 0 -1 0 1 0
T5 (keuangan) :	0 1 -1 0 0 0
T6 (valentino) :	1 -1 0 0 0 0
T7 (rossi) :	1 0 0 0 0 -1
T8 (tes) :	0 0 0 0 -1 1
T9 (sirkuit) :	1 0 -1 0 0 0

Dengan informasi jumlah dokumen dan *term* yang digunakan maka *term-document* sebelum direduksi berukuran 8 x 3. Matriks baru yang merupakan matriks yang telah direduksi dimensinya berukuran 6 x 3 dengan setiap vektor dokumen memiliki nilai sebagai berikut:

D1 :	1 2 -1 -2 1 -1
D2 :	0 2 -1 -1 1 -1
D3 :	3 -1 -1 0 -1 0

Sebagai contoh, nilai D3 pada matriks diatas didapat dari penjumlahan vektor *term* T6(valentino), T7(rossi), T8(tes), dan T9(sirkuit) yang merupakan *term-term* yang muncul pada D3.

Secara matematis tahapan mereduksi *term-document matrix* M *m x n* menjadi matriks M *k x n* dengan *k* << *n*, *m* adalah jumlah *term,* dan *n* adalah jumlah dokumen dapat ditulis sebagai (Bingham & Mannila, 2001)

$$M_{k \times n} = Random_{k \times m} M_{m \times n} \tag{17}$$

Dari persamaan (17), $Random_{kxm}$ adalah matriks acak yang dibentuk pada tahap 2. Persamaan ini yang digunakan dalam implementasi Random Projection. Dari persamaan tersebut, dapat ditentukan kompleksitas dari teknik Random Projection yaitu pembentukan matriks acak *O(km)* dan perkalian matriks acak dengan *term-document matrix O(kmn)* (Fradkin & Madigan, 2002).

Ada banyak pilihan dalam mengkonstruksi matriks acak. Secara umum, distribusi elemen-elemen pada matriks mengikuti distribusi Gaussian. Jika tiap vektor dalam matriks acak saling orthogonal satu sama lain maka jarak antar vektor pada matriks asli akan dipertahankan 100%. Namun, hal ini memerlukan biaya yang cukup tinggi sehingga tiap vektor pada matriks acak tidak perlu 100% orthogonal tetapi mendekati orthogonal (Hecht-Nielsen, 1994). Ukuran mendekati orthogonal yang digunakan adalah *mean square* dari $Random^T Random$ dengan matriks identitas adalah *1/k* per elemen dimana *k* adalah besarnya dimensi baru (Bingham & Mannila, 2001).

Pada (Achlioptas, 2001) diusulkan distribusi yang lebih sederhana dan memberikan hasil yang cukup baik. Dua distribusi yang diusulkan adalah

$$Random_{i,j} = \begin{cases} +1, prob = 1/2 \\ -1, prob = 1/2 \end{cases} \quad (18)$$

$$Random_{i,j} = \sqrt{3} \cdot \begin{cases} +1, prob = 1/6 \\ 0, prob = 2/3 \\ -1, prob = 1/6 \end{cases} \quad (19)$$

2.7. K-Means

K-Means adalah algoritma pengelompokan (*clustering*) yang bersifat *unsupervised* dan *partitional* (Dubes & Jain, 1988). Dengan sifat tersebut, K-Means tidak memiliki informasi atau pengetahuan terlebih dahulu sebelum melakukan pengelompokan (*unsupervised*). Pengelompokan dilakukan berdasarkan struktur yang ada pada koleksi data. Data yang memiliki kemiripan struktur yang tinggi akan dikelompokkan ke dalam satu kluster. Sedangkan sifat *partitional* berarti bahwa K-

Means mengelompokkan data-data ke dalam kluster-kluster yang sejajar satu sama lain dan saling lepas (*disjoint*).

Jika diberikan sekumpulan data yang akan dikelompokkan dan jumlah kelompok yang ingin dibentuk, maka algoritma ini akan mempartisi kumpulan data tersebut ke dalam kluster yang saling lepas sejumlah kelompok yang ingin dibentuk. Ide dari algoritma ini adalah setiap data memiliki lokasinya tersendiri dalam suatu ruang. Data yang berdekatan dianggap memiliki kemiripan dan akan dikelompokkan ke dalam satu kelompok.

Cara kerja algoritma ini secara umum adalah untuk *k* buah kluster yang ingin dibentuk, terlebih dahulu didefinisikan *k* buah pusat kluster, satu untuk masing-masing kluster. Pemilihan pusat ini diusahakan yang memiliki jarak yang sejauh mungkin satu sama lain. Langkah selanjutnya adalah memasukkan setiap data ke dalam kluster yang pusatnya paling dekat dengan data yang sedang diobservasi. Setelah semua data dimasukkan ke dalam kluster, maka iterasi pertama selesai. Penentuan pusat kluster dilakukan kembali, pusat yang dipilih adalah yang memiliki jumlah jarak minimum dengan data lain dalam kluster yang bersangkutan. Setelah itu, penentuan kluster dilakukan kembali untuk tiap data. Penentuan pusat kluster dan penentuan kluster dilakukan sampai pusat kluster tidak berpindah lagi.

Objective function yang digunakan pada algoritma ini adalah

$$\min J = \min \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{20}$$

$\left\| x_i^{(j)} - c_j \right\|^2$ adalah besarnya jarak antara data $x_i^{(j)}$ dengan pusat kluster c_j.

Secara singkat algoritma K-Means dapat dituliskan sebagai berikut:

1. Pilih K buah pusat kluster.
2. Masukkan setiap data ke kluster yang pusat nya paling dekat.
3. Setelah semua data sudah dimasukkan ke dalam kluster, hitung ulang K buah pusat kluster.
4. Ulangi langkah 2 dan 3 sampai pusat kluster tidak berpindah lagi.