Dekomposisi Matriks dalam Latent Semantic Indexing (LSI) untuk Pemrosesan Dokumen

Muhammad Iqbal Haidar - 13523111[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13523111@std.stei.itb.ac.id, [1]iqbalhaidar0603@gmail.com

***Abstract*—Seiring dengan perkembangan zaman, kebutuhan untuk pencarian informasi semakin tinggi. Sebuah metode andal diperlukan agar proses pencarian informasi dapat berjalan dengan efektif dan efisien. Latent Semantic Indexing hadir untuk menjawab tantangan tersebut. Dengan memanfaatkan penerapan teori dekomposisi matriks *Single Value Decomposition* dapat dibuat sebuah sistem temu balik sederhana dari sebuah kumpulan dokumen. Cukup dengan memasukkan query yang relevan maka komputer akan mencari dokumen tersebut berdasarkan kesamaan menggunakan *cosine similarity*.**

***Keywords*— Dokumen, Indexing, Single Value Decomposition, Temu Balik.**

I. PENDAHULUAN

Di era digital seperti saat ini, sudah tak terhitung banyaknya informasi penting yang didokumentasikan secara digital dalam sebuah *file* dan disimpan pada media penyimpanan elektronik. Dokumentasi fisik menggunakan kertas sudah mulai ditinggalkan karena dianggap rumit dan tidak ramah lingkungan. Banyak sekali dokumen baik yang bersifat penting ataupun trivial sudah tersedia dalam bentuk elektroniknya mulai dari sertifikat digital sampai catatan belanja sehari-hari. Salah satu keuntungan dari sebuah dokumen elektronik adalah kemudahan untuk berbagi dengan orang lain serta tidak perlu untuk membawa-bawa dokumen yang berat, cukup mengaksesnya menggunakan gawai pintar di genggaman tangan. Ini merupakan hal positif yang harus terus diadaptasi dimasa depan untuk menyokong umat manusia berakselerasi menghadapi perkembangan zaman.

Dahulu sebuah dokumen yang sudah tidak lagi digunakan akan disimpan dalam sebuah ruangan arsip yang diatur oleh arsiparis. Hal ini diperuntukkan supaya apabila dimasa yang akan datang kiranya diperlukan kembali untuk meninjau dokumen tersebut maka cukup dengan mengambilnya dari ruangan arsip. Selain itu dokumen yang dijadikan arsip juga dapat digunakan sebagai bukti yang kuat untuk pencatatan sebuah perjanjian ataupun kontrak apabila terjadi sengketa. Seorang arsiparis biasa menyimpan dokumen dalam keterurutan tertentu seperti berdasarkan tanggal pembuatan dokumen atau huruf abjad. Namun hal itu bisa dirasa kurang efisien sehingga muncul sebuah metode penyimpanan arsip dengan pengelompokan berdasarkan konteks dokumen sehingga proses pencarian dokumen dirasa lebih efektif.

Sebuah dokumen yang disimpan dalam bentuk elektronik sebenarnya tidak bisa dilakukan pengarsipan seperti layaknya dokumen fisik sebab data yang disimpan dalam media penyimpanan seperti *harddisk* hanya menyimpan informasi dalam bentuk biner. Sehingga semua dokumen yang tersimpan di media penyimpanan tersebut menjadi satu tanpa ada pemisahan tertentu, kalaupun ada itu hanya secara abstraksi bukan fisik. Akibat dari permasalahan tersebut adalah sulitnya bagi pengguna/pemilik dokumen untuk mencari dokumen yang sudah diarsipkan, harus dilakukan iterasi satu persatu melihat isi dari setiap dokumen yang disimpan. Hal ini dinilai kurang efektif sehingga diperlukan sebuah solusi yang mampu menampilkan dokumen-dokumen relevan dengan melakukan *query* pencarian berdasarkan kata kunci sebagai konteks dari dokumen yang ingin dicari.

Latent Semantic Indexing (LSI) hadir untuk menjawab tantangan tersebut, ini merupakan metode sebuah pencatatan dokumen yang tersimpan secara elektronik. *LSI* memungkinkan pengguna untuk mencari dokumen-dokumen elektronik pada komputer dengan cukup memasukkan *query* kata kunci ataupun kalimat yang nantinya akan dijadikan sebagai acuan konteks untuk menemukan dokumen yang relevan. Prinsip kerja dari metode ini adalah dengan meninjau seluruh struktur kata-kata yang ada dalam sebuah dokumen dan akan disimpan dalam sebuah basis data lalu nantinya data ini seolah-olah akan menjadi identitas dari dokumen tersebut. Sehingga apabila pengguna ingin mencari sebuah dokumen, maka cukup memasukkan sebuah *query* yang kiranya berkaitan dengan dokumen tersebut lalu komputer akan menghitung kemiripan antara *query* dengan identitas dokumen pada *database*. Keunggulan dari metode ini ketimbang yang lain adalah akurasi yang lebih tinggi sebab dapat menangani kasus sinonim dan polisemi. Metode ini tidak hanya mencari dokumen berdasarkan kuantitas jumlah kata yang paling sering muncul, tetapi juga mencoba untuk mengevaluasi konteks dan makna kata dalam dokumen.

Secara garis besar makalah ini akan membahas teori dibalik *Latent Semantic Indexing (LSI)* yang sudah umum digunakan untuk membantu manusia menemukan sebuah dokumen yang disimpan dalam sebuah media penyimpanan elektronik. *Latent Semantic Indexing (LSI)* memanfaatkan teori dalam Aljabar Linear yakni dekomposisi matriks menggunakan metode *Single Value Decomposition (SVD)* yang juga sudah umum diaplikasikan pada kehidupan sehari-hari untuk melakukan peninjauan serta pencarian identitas unik dari setiap

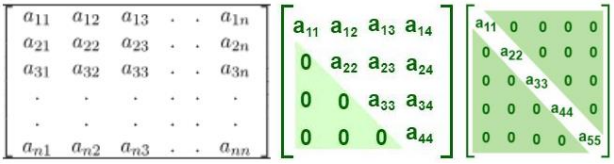
dokumen yang disimpan. Dengan mengimplementasikan *Latent Semantic Indexing (LSI)* ini maka masalah pencarian dokumen digital yang disimpan dalam media elektronik akan menjadi lebih mudah dan efisien sebab tidak perlu lagi untuk melakukan iterasi satu persatu melihat dokumen secara manual, cukup dengan memasukkan *query* yang akan dijadikan sebagai konteks maka komputer akan mencarikan dokumen yang relevan untuk kita.

II. LANDASAN TEORI

A. Matriks

Sebuah matriks didefinisikan sebagai kumpulan bilangan-bilangan baik itu bilangan riil ataupun bilangan kompleks yang tersusun sedemikian rupa sehingga berbentuk seperti tabel dengan ukuran $m \times n$ dimana m adalah jumlah baris yang dimiliki oleh matriks tersebut dan n adalah jumlah kolom yang dimiliki oleh matriks tersebut serta kedua ujungnya dibatasi dengan tanda kurung siku. Konsep matriks ini banyak diimplementasikan dalam kehidupan sehari-hari seperti untuk penyelesaian sebuah permasalahan sistem persamaan linear dan transformasi geometri. Ada banyak kasus khusus dimana matriks dikategorikan dengan ciri-ciri tertentu, beberapa diantaranya yang akan dibahas karena relevan dengan topik ini adalah matriks segitiga, matriks persegi, dan matriks diagonal.

Matriks persegi adalah matriks yang memiliki banyak baris dan banyak kolom yang sama sehingga ukuran matriks tersebut berupa $n \times n$ serta pada matriks jenis ini juga terdapat sebuah diagonal utama yakni elemen-elemen yang berada di garis diagonal dari kiri atas menuju kanan bawah. Matriks segitiga memiliki dua jenis berbeda yakni matriks segitiga atas dan matriks segitiga bawah. Matriks segitiga atas didefinisikan sebagai sebuah matriks dimana semua elemen di bawah diagonal utama matriks bernilai nol sedangkan matriks segitiga bawah memiliki definisi yang berkebalikan dengan matriks segitiga atas. Terakhir adalah matriks diagonal, yang memiliki pengertian sebuah matriks yang elemen selain elemen diagonalnya bernilai nol. Matriks diagonal ini digunakan pada metode *Singular Value Decomposition (SVD).*



Gambar 2.1 Matriks Persegi, Segitiga Atas, dan Diagonal
Sumber: https://www.geeksforgeeks.org/

Perkalian matriks dapat dilakukan untuk lebih dari dua matriks, dimana produk dari perkalian kedua matriks tersebut adalah sebuah matriks baru hasil perkalian. Apabila dua buah matriks berukuran $m_1 \times n_1$ dikalikan dengan $m_2 \times n_2$ maka matriks hasil perkaliannya berukuran $m_1 \times n_2$ dengan catatan besaran n_1 harus sama dengan m_2 dimana m dan n adalah representasi jumlah baris dan kolom untuk masing-masing matriks. Proses perkaliannya dilakukan dengan cara mengambil setiap elemen pada setiap baris dari matriks satu kemudian mengambil setiap elemen pada setiap kolom dari matriks dua lalu mengalikan setiap elemen yang berkorespondensi. Nilai elemen pada baris dan kolom di matriks hasil berupa penjumlahan hasil perkalian dari semua elemen pada baris dan kolom tersebut.

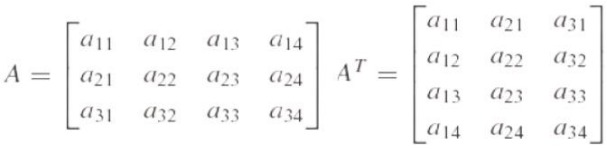
Determinan sebuah matriks didefinisikan sebagai hasil sebuah bilangan skalar yang merepresentasikan ciri dari matriks tersebut. Perlu diketahui bahwa determinan dari sebuah matriks hanya akan bisa dicari apabila matriks tersebut merupakan matriks persegi yakni banyak baris dan kolomnya memiliki besaran yang sama yakni $n \times n$. Apabila sebuah matriks bukan merupakan matriks persegi maka dapat dikatakan bahwa determinan dari matriks tersebut tidak dapat ditemukan atau tidak ada. Ada banyak metode yang dapat digunakan untuk mencari nilai determinan sebuah matriks persegi, beberapa diantaranya adalah dengan menggunakan ekspansi kofaktor, reduksi baris ataupun aturan sarrus untuk matriks dengan ukuran yang tidak terlalu besar. Nilai determinan ini banyak digunakan dalam teori lain pada Aljabar Linear seperti misalnya apabila sebuah matriks tidak dapat dicari nilai determinannya atau determinannya sama dengan nol maka dapat dipastikan bahwa matriks tersebut tidak memiliki *inverse*.

Sebuah *inverse* matriks didefinisikan sebagai sebuah matriks lain yang apabila matriks tersebut dikalikan dengan matriks yang dicari *inverse*nya maka akan menghasilkan sebuah matriks identitas. Matriks identitas adalah sebuah matriks yang memiliki ukuran yang sama seperti matriks *inverse*nya namun dengan setiap elemen pada diagonal utamanya bernilai angka satu dan elemen lainnya bernilai nol. Sama seperti pengertian *inverse* pada teori matematika lainnya, *inverse* matriks juga dapat dikatakan sebagai lawan dari matriks yang sebenarnya. Sebagai catatan, *inverse* dari sebuah matriks hanya akan bisa dicari apabila matriks tersebut memiliki determinan dan determinannya terdefinisi bukan nol. Untuk mencari *inverse* dari sebuah matriks dapat menggunakan metode *determinan-adjoin.*

$$\begin{bmatrix} 2 & -5 \\ -1 & 3 \end{bmatrix}\begin{bmatrix} 3 & 5 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Gambar 2.2 Matriks dan Inversenya
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

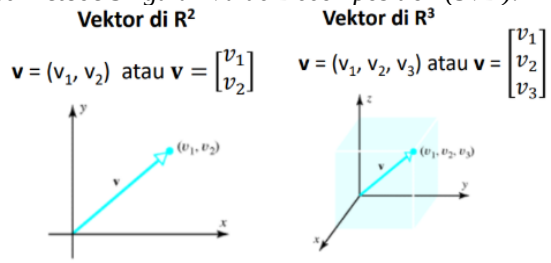
Transpose dari sebuah matriks didefinisikan sebagai matriks lain yang ukurannya sama namun elemennya ditukar sedemikian rupa sehingga setiap elemen pada baris dan kolom dari matriks awalan menjadi elemen pada kolom dan baris pada matriks *transpose*nya. Pada esensinya sebuah transpose matriks adalah penukaran elemen pada matriks awalan sehingga elemen ke *xy* menjadi elemen ke *yx* pada matriks transposenya. Apabila misalkan sebuah matriks berukuran $m \times n$ akan dicari transposenya maka akan menghasilkan sebuah matriks transpose dengan ukuran yakni $n \times m$. Matriks transpose ini berguna dalam implementasi *Singular Value Decomposition (SVD)* yang merupakan metode untuk mendekomposisi sebuah matriks menjadi beberapa perkalian matriks lainnya.



Gambar 2.3 Transpose Matriks
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

B. Vektor

Vektor didefinisikan sebagai sebuah besaran yang dia memiliki dua buah properti khas yang membedakannya dengan besaran pada umumnya yakni memiliki representasi kuantitas serta memiliki representasi arah. Sehingga bisa dikatakan bahwa vektor merupakan sebuah besaran yang memiliki arah. Sebuah vektor dapat dinotasikan dengan berbagai cara seperti berupa huruf kecil yang dicetak tebal ataupun huruf yang memiliki lambang panah pada bagian atasnya. Vektor juga memiliki sebuah ruang vektor yakni tempat dimana vektor tersebut didefinisikan, apabila sebuah vektor memiliki dimensi kurang dari sama dengan tiga maka vektor tersebut dapat digambarkan pada ruang *euclidean* atau ruang tiga dimensi dengan sumbu *xyz*. Aplikasi konsep vektor ini banyak sekali digunakan dalam kehidupan manusia sehari-hari seperti untuk menghitung kecepatan mobil menggunakan konsep vektor karena kecepatan memiliki kuantitas dan arah bergeraknya. Pada makalah ini konsep vektor yang digunakan merupakan perkembangan lebih lanjutnya yakni berupa *vektor eigen* atau vektor asli yang diaplikasikan pada metode *Singular Value Decomposition (SVD)*.

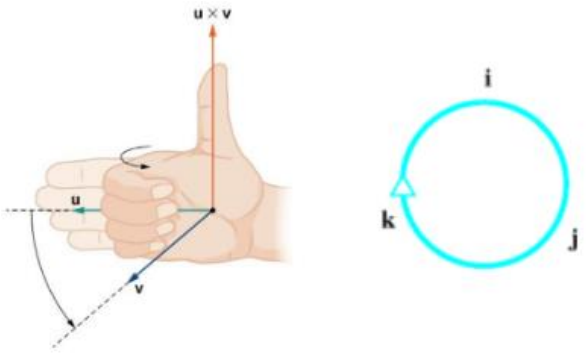


Gambar 2.4 Vektor
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

Definisi dari sebuah vektor tidak nol adalah sebuah vektor yang komponen-komponen penyusunnya tidak terdiri hanya atas bilangan nol saja. Akibat dari pendefinisian tersebut maka dapat dipastikan sebuah vektor tersebut akan memiliki panjang atau norma vektor tidak berupa nol sebab dipastikan ada bilangan lain yang terdefinisi bukan nol pada komponen penyusunnya. Penggunaan vektor tak nol ini banyak digunakan pada teori yang dipelajari dalam Aljabar Linear salah satunya adalah untuk sebagai salah satu dasar dalam pembentukan ruang vektor yang mendefinisikan sebuah vektor dengan menggunakan metode kombinasi linear yakni membentuk sebuah vektor melalui operasi matematika vektor-vektor lainnya.

Sebuah vektor dapat memiliki properti operasi matematika yakni perkalian. Dalam vektor terdapat dua jenis perkalian yang menghasilkan produk berbeda sesuai dengan operasinya. Operasi perkalian yang pertama pada vektor adalah perkalian *dot* atau biasa disebut perkalian titik dimana hasil dari perkalian dua buah vektor tersebut adalah sebuah skalar sehingga dari awalnya berupa vektor yang memiliki properti arah akan nantinya menjadi skalar yang tidak memiliki properti arah, hanya kuantitas saja. Prosedur untuk melakukan perkalian *dot* atau titik pada dua buah vektor adalah cukup dengan mengalikan setiap komponen dari kedua vektor yang berkoresponden kemudian menjumlahkan hasil perkalian tersebut. Hasil penjumlahan tersebutlah yang dinamakan produk hasil perkalian titik dari kedua vektor. Perlu dicatat bahwa apabila kedua vektor saling tegak lurus satu dengan lainnya maka dapat dipastikan bahwa hasil perkalian titik dari kedua vektor tersebut bernilai nol. Aplikasi perkalian titik banyak digunakan dalam ilmu fisika misalnya untuk menghitung kerja pada sebuah objek digunakan perkalian vektor yakni vektor gaya dan vektor perpindahan.

Jenis perkalian vektor lainnya adalah perkalian *cross* atau biasa disebut perkalian silang, dimana hasil dari perkalian dua buah vektor menggunakan jenis perkalian vektor ini adalah vektor baru sehingga masih memiliki dua properti yakni kuantitas dan arah. Prosedur untuk melakukan operasi perkalian ini terbilang cukup lebih rumit ketimbang perkalian dua vektor menggunakan perkalian titik sebab harus menghitung arah dari hasil perkalian tiap elemennya. Cara menghitung perkalian *cross* dari dua vektor yang berbeda adalah dengan mengalikan setiap elemen pada vektor satu dengan setiap elemen pada vektor dua lalu setelah ditemukan hasil kuantitasnya akan dicari arahnya berdasarkan arah kedua elemen awal tersebut menggunakan aturan siklus ataupun aturan tangan kanan. Perlu dicatat bahwa jika dua buah vektor yang dikalikan menggunakan perkalian *cross* saling sejajar atau sudut yang dibentuk antara keduanya berupa 180 derajat maka dapat dipastikan bahwa hasil perkalian *cross* dari vektor tersebut adalah nol. Perkalian *cross* ini banyak digunakan dalam bidang fisika, salah satunya adalah untuk menghitung torsi yang dihasilkan dengan mengalikan vektor posisi dengan vektor gaya.



Gambar 2.5 Aturan Tangan Kanan dan Siklus
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

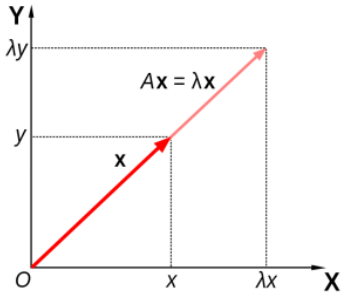
Panjang vektor atau bisa juga disebut dengan *norma/magnitudo* dapat didefinisikan sebagai representasi vektor dalam skalarnya atau bisa dikatakan sebuah ukuran kuantitas dari sebuah vektor dengan mengabaikan properti arahnya dalam sebuah ruang vektor tempat vektor tersebut didefinisikan. Sebuah vektor nol dimana setiap komponen penyusun vektor tersebut bernilai nol maka akan mengakibatkan panjang vektornya juga bernilai nol. Untuk

menghitung panjang sebuah vektor dapat menggunakan rumus yang sudah ditemukan dan ditetapkan sebelumnya. Ada juga konsep berupa vektor satuan yakni dimana komponen penyusun vektor satuan tersebut apabila dicari panjang vektornya akan menghasilkan nilai berupa satu. Aplikasi konsep panjang vektor ini banyak digunakan pada berbagai bidang ilmu pengetahuan salah satunya adalah ilmu komputer, dimana panjang vektor digunakan untuk menganalisis kesamaan pola yang digunakan untuk penelitian pembelajaran mesin.

Ortogonal dan ortonormal sebenarnya memiliki definisi yang serupa namun tidak sama. Apabila misalkan terdapat dua buah vektor yang jika kedua vektor tersebut digambarkan pada ruang vektornya akan saling tegak lurus atau membentuk sudut 90 derajat maka dapat dikatakan bahwa vektor tersebut ortogonal atau saling tegak lurus. Ortonormal memiliki pengertian yang mirip dengan ortogonal namun dengan tambahan syarat berupa kedua vektor yang ortogonal tersebut semuanya memiliki panjang vektor bernilai satu atau merupakan vektor satuan. Oleh karena itu dapat dikatakan bahwa ortonormal merupakan *subset* atau kasus khusus dari ortogonal, dengan demikian apabila dua vektor dikatakan ortonormal maka dapat dipastikan keduanya juga ortogonal. Untuk memeriksa apakah dua buah vektor dapat dikatakan ortogonal/ortonormal dapat dilakukan dengan perkalian dot kedua vektor, apabila hasil dari perkalian tersebut berupa nol maka dapat dikatakan kedua vektor ortogonal dan apabila panjang dari kedua vektor tersebut juga bernilai satu maka vektor tersebut dapat dikatakan ortonormal. lainnya.

C. Vektor Eigen

Vektor *eigen* berasal dari kata dalam bahasa jerman yakni *eigen* yang berarti asli atau bisa juga diinterpretasikan sebagai karakteristik sehingga dapat dikatakan bahwa vektor *eigen* merepresentasikan nilai karakteristik dari sebuah matriks persegi berukuran $n \times n$. Pendefinisian vektor eigen adalah apabila terdapat sebuah matriks persegi katakanlah A dimana banyak baris dan kolomnya sama berukuran $n \times n$ maka akan ditemukan sebuah vektor x pada ruang vektor n sedemikian sehingga $Ax = \lambda x$ serta x dapat dikatakan sebagai vektor *eigen* yang berkorespondensi dengan *lambda*. *Lambda* adalah sebuah konstanta yang disebut nilai *eigen*, *lambda* ini merupakan faktor yang dapat mengalikan vektor *eigen* x sehingga memanjang ataupun memendek bahkan berbalik arah.



Gambar 2.6 Vektor Eigen
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

Untuk melakukan pencarian sebuah vektor *eigen* dari matriks persegi $n \times n$ dapat menggunakan rumus turunan yang sudah ditemukan oleh para ahli yakni $(\lambda - A)x = 0$ dan $\det(\lambda - A) = 0$ Dengan menggunakan rumus persamaan di samping yang juga memiliki nama lain yakni persamaan karakteristik maka akan ditemukan nilai akar-akar dari persamaan karakteristik tersebut yakni *lambda* yang akan menjadi nilai *eigen* dari matriks persegi yang dicari nilai *eigen*nya dalam hal ini adalah matriks A. Pada prosesnya penghitungan nilai *eigen*nya, juga akan bisa dapat ditemukan vektor *eigen* serta ruang *eigen* yang membangun vektor tersebut. Terdapat teorema terkait hubungan nilai *eigen* ini dengan apakah matriks A memiliki balikan *(inverse)*. Dikatakan bahwa sebuah matriks persegi A dimana ukurannya berupa $n \times n$ akan dapat ditemukan *inverse*nya jika dan hanya jika nilai *eigen*nya tidak ada yang bernilai nol.

Sebuah matriks persegi misalkan A yang berukuran $n \times n$ dapat dikatakan bisa untuk dilakukan diagonalisasi apabila matriks tersebut mirip seperti matriks diagonal yang elemen-elemen selain pada diagonal utamanya bernilai nol. Persamaan umum yang digunakan adalah, terdapat sebuah matriks P sedemikian sehingga $P^{-1}AP$ adalah sebuah matriks diagonal. Apabila kondisi tersebut terpenuhi maka dapat dikatakan dalam hal ini bahwa matriks P telah mendiagonalisasi matriks persegi $n \times n$ A. Dimana P adalah sebuah matriks yang kolom-kolomnya berupa basis dari ruang *eigen* yang telah ditemukan dari matriks A. Teknik diagonalisasi ini berguna untuk menghitung perkalian matriks dengan ukuran/orde yang tinggi secara berulang-ulang.

$$D = P^{-1}AP$$
$$= \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 1 \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -2 \\ 1 & 2 & 1 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} -1 & 0 & -2 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

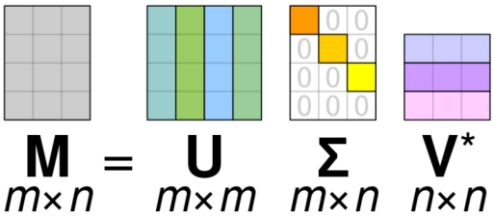
Gambar 2.7 Diagonalisasi Vektor
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

III. SINGULAR VALUE DECOMPOSITION (SVD)

Sebuah matriks dapat dilakukan proses dekomposisi yakni memfaktorkan matriks tersebut sehingga hasil perkalian dari komponen-komponen hasil dekomposisi menghasilkan matriks awal. Dengan kata lain, dekomposisi matriks merupakan proses memecahkan matriks menjadi matriks-matriks lainnya yang apabila keseluruhan matriks tersebut dikalikan akan menghasilkan matriks akhir yang sama seperti matriks awal sebelum dekomposisi dilaksanakan. Ada beberapa teknik dekomposisi yang tersedia dan dapat digunakan seperti misalnya teknik dekomposisi *LU* yang memecahkan matriks menjadi matriks segitiga, kemudian ada dekomposisi *QR* yang mendekomposisi matriks menjadi matriks ortogonal serta segitiga atas, namun yang akan dibahas serta digunakan pada metode *Latent Semantic Indexing (LSI)* adalah dekomposisi *Singular Value Decomposition (SVD)*.

Singular Value Decomposition (SVD) merupakan salah

satu teknik dekomposisi matriks yang mampu memfaktorkan sebuah matriks non persegi $m \times n$ yang dimisalkan dengan A. *SVD* memfaktorkan matriks awal menjadi matriks U, Σ, V sedemikian sehingga memenuhi persamaan $A = U\Sigma V^T$ dimana U merupakan matriks ortogonal (vektor singular kiri) yang berukuran $m \times m$ dan $n \times n$ merupakan matriks ortogonal (vektor singular kanan) yang berukuran $n \times n$ serta sigma adalah sebuah matriks diagonal dimana matriks tersebut berukuran $m \times n$ elemen serta elemen pada diagonal utamanya adalah nilai-nilai singular dari matriks A dan sisa elemen lainnya bernilai nol. Perlu menjadi catatan bahwa matriks ortogonal didefinisikan sebagai matriks dengan setiap kolom-kolomnya berupa vektor yang ortogonal. Diagonal utama pada matriks non-persegi adalah garis yang ditarik dari kiri atas menuju sejauh mungkin ke kanan bawah. Nilai singular matriks dinotasikan dengan σ serta didefinisikan sebagai akar dari nilai-nilai *eigen* pada matriks A.



Gambar 3.1 Singular Value Decomposition (SVD)
Sumber:
https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/

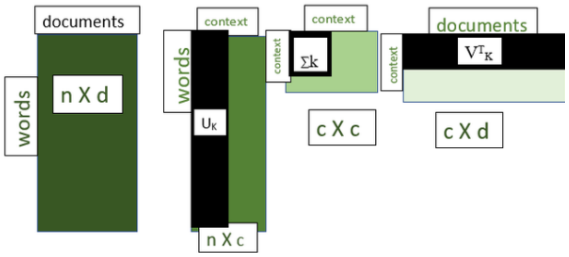
Terdapat dua cara untuk melakukan dekomposisi *Single Value Decomposition (SVD)* yakni menggunakan teorema dan menggunakan metode perhitungan terpisah vektor singular kiri dan vektor singular kanan, namun pada makalah ini hanya akan dibahas cara menghitung dekomposisi menggunakan teorema. Dengan menggunakan metode teorema, kurang lebih terdapat lima langkah yang harus dilakukan untuk menemukan dekomposisi *SVD* dari sebuah matriks A yakni,

1. Tentukan vektor singular kanan V dengan mencari nilai-nilai *eigen* yang berkorespondensi dengan matriks A kemudian lakukan normalisasi dan transpose kan matriks sehingga menjadi V^T
2. Tentukan vektor singular kiri U dengan menggunakan persamaan di bawah ini, tidak lupa lakukan normalisasi terlebih dahulu untuk semua vektor.
3. Apabila $n > k$, maka perluaslah vektor U untuk membentuk sebuah basis ortonormal untuk ruang vektor berdimensi n
4. Tentukan matriks Σ berukuran baris dan kolom yang identik seperti matriks A dengan elemen diagonal utamanya adalah nilai-nilai singular yang tidak nol dari matriks A yang tersusun dengan urutan besar menuju kecil.
5. Komponen dekomposisi matriks A dapat ditemukan.

IV. LATENT SEMANTIC INDEXING (LSI)

Latent Semantic Indexing (LSI) merupakan sebuah metode yang biasanya digunakan dalam studi bidang ilmu *Natural Language Processing* serta *Infomation Retrieval* yang berfungsi untuk mencari data-data berdasarkan masukkan *query* dari pengguna. *Latent Semantic Indexing (LSI)* melakukan peninjauan terhadap dokumen-dokumen menumpuk yang biasanya menjadi database atau acuan untuk melakukan Temu Balik Informasi. *LSI* akan melakukan peninjauan terhadap hubungan makna-makna yang ada di dalam dokumen tersebut lalu hasil dari analisis akan dijadikan sebagai "identitas" yang dimiliki oleh dokumen tersebut. Apabila seorang pengguna ingin mencari sebuah dokumen maka cukup memasukkan *query* yang sekiranya relevan dengan dokumen yang diinginkan, lalu komputer akan menganalisis *query* tersebut dan melakukan proses pencarian/*matching* terhadap dokumen yang tersedia dari database. Komputer kemudian menampilkan dokumen-dokumen yang dinilai mirip dengan *query* masukkan dari pengguna. Implementasi *LSI* ini tentunya memberikan dampak positif yang cukup besar dalam kehidupan manusia sehari-hari.

Urutan proses dari *Latent Semantic Indexing (LSI)* yang paling pertama adalah melakukan pencarian dokumen-dokumen yang nantinya akan dijadikan sebagai *database* pencarian. Kemudian dokumen-dokumen tersebut akan dilakukan *pre-processing text* yakni menghilangkan kata-kata yang kurang bermakna seperti "dan", "atau", "adalah" serta mengubah seluruh kata bentukan menjadi kata dasar. Selanjutnya akan dilakukan pembuatan matriks *term* serta penimbangan *term* menggunakan metode *TF-IDF* untuk meningkatkan akurasi dari *database*. Dari matriks yang sudah diproses sebelumnya akan dilakukan dekomposisi dengan metode *Singular Value Decomposition (SVD)* serta mereduksi dimensi guna mengurangi *noise* yang ada serta menurunkan kompleksitas perhitungan dengan mengambil beberapa nilai singular teratas. Terakhir pengguna dapat memasukkan *query* kemudian program akan mencari kemiripan *query* tersebut dengan dokumen yang tersedia di dalam *database*.



Gambar 4.1 Latent Semantic Indexing (LSI)
Sumber: https://www.geeksforgeeks.org/latent-semantic-analysis/

V. KESIMPULAN

Berdasarkan pembahasan di atas dapat ditarik kesimpulan bahwa dekomposisi matriks khususnya dengan metode *Singular Value Decomposition (SVD)* dapat diimplementasikan untuk memecahkan permasalahan sehari-hari seperti dalam *Latent Semantic Indexing (LSI)*. Penggunaan *LSI* ini dapat mempermudah serta membuat kerja manusia menjadi lebih efisien sehingga waktu tidak habis digunakan untuk hal-hal yang bersifat repetitif.

Bukan tidak mungkin, dimasa yang akan datang bermunculan implementasi-implementasi lain yang memanfaatkan teori-teori dari Aljabar Linear untuk memecahkan persoalan khususnya dibidang komputer.

UCAPAN TERIMA KASIH

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas berkat, rahmat, dan hidayat-Nya sehingga penulisan makalah ini bisa terselesaikan. Penulis juga ingin menyampaikan ucapan terima kasih kepada orang tua yang senantiasa memberikan semangat dan dukungan selama proses pengerjaan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada Bapak Ir. Rila Mandala, M.Eng., Ph.D. selaku dosen pengampu mata kuliah matematika diskrit (IF1220) kelas K-1 serta Bapak Dr. Ir. Rinaldi Munir, M.T., yang telah menjadi pembimbing, memberikan ilmu dan pengetahuan terkait materi sebagai modal dalam mengembangkan makalah ini. Ucapan terima kasih juga disampaikan kepada semua rekan dan pihak lain yang turut mendukung proses penulisan makalah ini, baik secara moral maupun material, walaupun tidak dapat disebutkan satu per satu.

REFERENSI

[1] Diakses pada tanggal 28 Desember 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-01-Review-Matriks-2023.pdf

[2] Diakses pada tanggal 28 Desember 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/Algeo-11-Vektor-di-Ruang-Euclidean-Bag1-2024.pdf

[3] Diakses pada tanggal 28 Desember 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-13-Vektor-di-Ruang-Euclidean-Bag3-2023.pdf

[4] Diakses pada tanggal 28 Desember 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian1-2023.pdf

[5] Diakses pada tanggal 28 Desember 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-20-Nilai-Eigen-dan-Vektor-Eigen-Bagian2-2023.pdf

[6] Diakses pada tanggal 28 Desember 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf

[7] Diakses pada tanggal 28 Desember 2024. https://www.geeksforgeeks.org/diagonal-matrix/

[8] Diakses pada tanggal 28 Desember 2024. https://www.geeksforgeeks.org/triangular-matrix/

[9] Diakses pada tanggal 28 Desember 2024. https://www.geeksforgeeks.org/square-matrix/

[10] Diakses pada tanggal 28 Desember 2024. https://www.geeksforgeeks.org/latent-semantic-analysis/

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bekasi, 29 Desember 2024

Muhammad Iqbal Haidar - 13523111