

Penerapan Metode *Nonnegative Matrix Factorization* (NMF) dalam Rekomendasi Penjualan Barang di *e-commerce* Berdasarkan *Rating* yang Diberikan Pengguna

Muhammad Fakhry Zaki - 13524087

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

Email : 13524087@std.stei.itb.ac.id, mfakhry.zaki@gmail.com

Abstract—*E-commerce is one of many sector that has been on the rise in this era of information and technology. One important aspect in the e-commerce process is the process of giving recommendation in selling things to the users. Nonnegative Matrix Factorization (NMF) is one of the methods that can be used to do such thing. This method factorizes a nonnegative matrix into two smaller matrix that is also nonnegative. The program analysis results show that this method is still far from perfection having many deficiency in giving users their recommendation. Despite that imperfection, this method is one of the methods that can be easily implemented using concepts of linier and geometric algebra while still being able to give relevant recommendation to users based on the ratings that they give to products bought by them.*

Keywords—*E-commerce, Factorization, Rating, Recommendation, User*

Abstrak—*E-commerce merupakan salah satu sektor yang berkembang dengan sangat pesat di tengah-tengah perkembangan teknologi informasi yang sangat marak ini. Salah satu aspek yang cukup penting dalam proses e-commerce ini adalah pemberian rekomendasi penjualan barang-barang kepada pengguna e-commerce tersebut. Salah satu metode yang bisa digunakan untuk pemberian rekomendasi tersebut adalah metode faktorisasi matriks nonnegatif. Faktorisasi ini memecah sebuah matriks nonnegatif menjadi dua buah matriks berdimensi lebih rendah yang juga bernilai nonnegatif. Hasil analisis program menunjukkan bahwa metode ini belum sempurna sebagai metode pemberian rekomendasi penjualan barang kepada pengguna. Meskipun demikian, metode ini merupakan salah satu metode yang terhitung mudah untuk diimplementasikan menggunakan konsep-konsep aljabar linier dan geometri serta tetap bisa memberikan rekomendasi yang cukup relevan kepada pengguna berdasarkan rating yang ia berikan pada produk-produk yang telah ia beli.*

Kata kunci—*E-commerce, Faktorisasi, Rating, Rekomendasi, Pengguna*

I. PENDAHULUAN

Perkembangan teknologi informasi yang sangat pesat menunjang pertumbuhan *e-commerce* dengan sangat pesat,

terutama dalam beberapa tahun terakhir ini. Saat ini *e-commerce* sudah menjadi bagian yang tidak terpisahkan dari kehidupan masyarakat di seluruh dunia. Kemudahan akses terhadap produk-produk yang diinginkan dengan adanya *e-commerce* ini menjadi salah satu alasan utama bertambahnya minat masyarakat dalam penggunaan aplikasi-aplikasi *e-commerce*.

Besarnya minat masyarakat dalam penggunaan *e-commerce* ini, tentu memperbesar pula pasar penjualan *e-commerce*. Hal ini terkadang bisa membuat pengguna kesulitan mencari barang-barang yang dibutuhkan. Oleh karena itu, diperlukan sebuah algoritma untuk pemberian rekomendasi barang-barang kepada pengguna berdasarkan hal-hal yang kemungkinan diinginkan oleh pengguna tersebut. Pemberian rekomendasi ini bisa diimplementasikan dengan berbagai cara, seperti dengan pengolahan riwayat pembelian pengguna, riwayat *rating* produk, dan lain sebagainya.

Salah satu algoritma pemberian rekomendasi yang mudah diimplementasikan dengan menerapkan konsep aljabar linier dan geometri adalah rekomendasi berbasis *rating* menggunakan metode *nonnegative matrix factorization* (NMF) atau faktorisasi matriks nonnegatif. Metode ini bekerja dengan memfaktorisasi matriks *rating* pengguna yang nilainya nonnegatif untuk mendapatkan pola laten yang merepresentasikan hubungan antara pengguna dengan produk yang ia beri *rating*. Pada makalah ini, akan dibahas contoh kecil simulasi penggunaan algoritma NMF untuk memberikan rekomendasi barang kepada pengguna berdasarkan *rating* yang ia berikan pada beberapa barang yang telah ia beli.



Gambar 1.1. Contoh-contoh *e-commerce* yang ada di Indonesia

II. DASAR TEORI

A. Matriks

Matriks adalah susunan elemen-elemen yang disusun dalam bentuk persegi panjang. Matriks terdiri dari dua dimensi, yakni baris dan kolom. Matriks biasanya dituliskan dengan kurung siku besar menyelimuti semua elemen-elemennya. Matriks digunakan untuk berbagai hal terutama dalam aljabar linier, di antaranya adalah penyelesaian sistem persamaan linier dan pemrosesan gambar.

B. Dekomposisi Matriks

Dekomposisi atau faktorisasi matriks adalah metode untuk memecah sebuah matriks menjadi dua atau lebih matriks lain yang lebih sederhana. Dengan melakukan dekomposisi pada sebuah matriks, struktur, pola, serta informasi penting yang terkandung pada matriks tersebut bisa lebih mudah diekstraksi. Dengan memecah matriks menjadi lebih sederhana, baik secara dimensi maupun secara nilai, operasi-operasi matematika bisa dilakukan dengan lebih mudah dan efisien tanpa kehilangan informasi penting yang terkandung pada matriks tersebut.

C. Jenis-jenis Dekomposisi Matriks

Terdapat banyak jenis dekomposisi matriks yang masing-masing memiliki fungsi yang berbeda-beda. Beberapa contoh dekomposisi matriks antara lain:

1. Dekomposisi SVD

Dekomposisi SVD (*Singular Value Decomposition*) adalah teknik dekomposisi matriks untuk menguraikan sebuah matriks menjadi tiga buah komponen, yakni

$$A = U\Sigma V^T$$

A = Matriks acak berukuran $m \times n$

U = Matriks ortogonal berukuran $m \times m$

Σ = Matriks berisi nilai-nilai singular dari A berukuran $m \times n$

V = Matriks ortogonal $n \times n$

Salah satu keunggulan dekomposisi matriks SVD adalah pada kemampuannya untuk melakukan modifikasi pada komponen matriks tanpa kehilangan kualitas data yang signifikan

2. Dekomposisi QR

Dekomposisi QR mendekomposisi matriks menjadi matriks ortogonal dan matriks segitiga atas.

$$A = QR$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} | & | & | \\ q_1 & q_2 & q_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}$$

A = Matriks acak berukuran $m \times n$

Q = Matriks ortogonal berukuran $m \times n$

R = Matriks segitiga atas berukuran $n \times n$

Salah satu keunggulan dekomposisi ini adalah dalam mempertahankan kestabilan hasil komputasi sehingga menjadi pilihan yang cocok untuk perhitungan-perhitungan yang memerlukan keakuratan tinggi.

3. Dekomposisi LU

Metode dekomposisi LU memfaktorkan sebuah matriks menjadi matriks segitiga atas dan matriks segitiga bawah.

$$A = LU$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

A = Matriks acak berukuran $n \times n$

L = Matriks segitiga bawah berukuran $n \times n$

U = Matriks segitiga atas berukuran $n \times n$

Untuk dekomposisi LU, kelebihan dekomposisi ini adalah pada kemampuannya untuk menjaga fitur-fitur visual yang tahan terhadap gangguan-gangguan data gambar seperti kompresi.

D. Norma Matriks

Sebagaimana vektor, matriks juga memiliki norma yang bisa dihitung. Salah satu cara penghitungan norma matriks adalah dengan menggunakan norma frobenius. Pada dasarnya, penghitungan norma frobenius pada matriks hampir sama dengan penghitungan jarak euclidean pada vektor. Pada sebuah matriks A berukuran $m \times n$, nilai norma frobeniusnya dapat dihitung sebagai berikut:

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (a_{ij})^2}$$

E. Faktorisasi Matriks Nonnegatif

Faktorisasi Matriks Nonnegatif adalah metode dekomposisi atau faktorisasi sebuah matriks V nonnegatif menjadi matriks W dan H yang nonnegatif pula.

$$V \approx WH$$

V = Matriks acak nonnegatif berukuran $m \times n$

W = Matriks nonnegatif berukuran $m \times k$

H = Matriks nonnegatif berukuran $k \times n$
Metode ini disebut juga sebagai aproksimasi matriks nonnegatif karena sejatinya metode ini tidak benar-benar memfaktorisasi matriks sebagaimana SVD, LU, dan QR melainkan hanya membuat aproksimasi atau pendekatan saja.

Kegunaan metode ini adalah untuk memetakan matriks V ke ruang laten yang berdimensi lebih kecil (k) sehingga aproksimasi pada matriks W bisa menjadi basis untuk V . Oleh karena itu, metode ini bisa digunakan untuk pemberian rekomendasi sederhana. Tujuan akhir dari implementasi metode faktorisasi matriks nonnegatif ini adalah untuk meminimalisir norma frobenius dari $V - WH$.

Aturan perhitungan untuk mendapatkan aproksimasi matriks W dan H yang baik mengikuti aturan iteratif yang konvergen. Implementasi perhitungan yang menyeimbangkan antara kecepatan dan kemudahan implementasi menurut *Lee* dan *Seung* pada mulanya matriks W dan H diinisialisasi dengan matriks acak, kemudian matriks W dan H di-update dengan aturan sebagai berikut:

$$H_{ij} = H_{ij} * \frac{(W^T V)_{ij}}{(W^T W H)_{ij}}$$

$$W_{ij} = W_{ij} * \frac{(V H^T)_{ij}}{(W H H^T)_{ij}}$$

III. LANGKAH-LANGKAH IMPLEMENTASI

Pada makalah ini, penerapan penggunaan metode faktorisasi matriks nonnegatif dilakukan dengan pembuatan program pemberian rekomendasi penjualan kategori barang pada suatu pengguna di sebuah *e-commerce*. Pemberian rekomendasi ini didasari pada *rating* yang diberikan pengguna tersebut dan juga *rating* yang diberikan pengguna-pengguna yang lainnya. Implementasi pembuatan program tersebut akan dijabarkan dengan langkah-langkah berikut:

A. Memilih Dataset

Langkah pertama yang dilakukan adalah memilih dataset yang akan digunakan untuk analisis. Dataset yang digunakan untuk makalah ini merupakan data *rating* pengguna terhadap produk-produk yang telah mereka beli pada suatu *e-commerce*. Dataset diperoleh dari website Kaggle, yang merupakan salah satu platform penyedia dataset publik yang dapat digunakan untuk keperluan penelitian dan pengembangan. Secara spesifik, dataset yang digunakan dalam implementasi program pada makalah ini diambil dari link berikut [dataset](#).

B. Membangun Matriks User-Item Rating

Matriks *User-Item Rating* dibuat dari dataset *rating e-commerce* dengan baris matriks merepresentasikan pengguna, kolom matriks merepresentasikan kategori barang yang dibeli user tersebut, serta sel (nilai) pada matriks

merepresentasikan *rating* pengguna pada baris tersebut terhadap kategori barang pada kolom tersebut.

Ukuran dimensi dari Matriks *User-Item Rating* ini kemudian disimpan dalam variabel *row* dan *col* yang kemudian akan digunakan untuk menjadi ukuran matriks W dan H nantinya.

```
folder_src = Path(__file__).parent
def readCSVtoMat():
    path_review = folder_src.parent / "data" / "reviews.csv"
    path_product = folder_src.parent / "data" / "products.csv"

    pd_review = pd.read_csv(path_review)
    pd_product = pd.read_csv(path_product)

    product_category = pd_product['category'].values

    review_reviewID = pd_review['review_id'].values
    review_userID = pd_review['user_id'].values
    review_product = pd_review['product_id'].values
    review_rating = pd_review['rating'].values

    # ubah product_id jadi category
    for i in range(len(review_product)):
        temp = review_product[i][1:]
        temp = int(temp)
        review_product[i] = product_category[temp-1]

    rows = np.unique(review_userID)
    cols = np.unique(review_product)
    resMatrix = np.zeros((len(rows), len(cols)))

    user_to_idx = {user: i for i, user in enumerate(rows)}
    prod_to_idx = {prod: j for j, prod in enumerate(cols)}

    # bikin matriks dari ratingnya
    for i in range(len(review_reviewID)):
        rowId = user_to_idx[review_userID[i]]
        colId = prod_to_idx[review_product[i]]

        if resMatrix[rowId][colId] == 0:
            resMatrix[rowId][colId] = review_rating[i]

    return resMatrix, cols
```

Gambar 3.1. Subprogram untuk membangun Matriks *User-Item Rating* dari dataset

Dari hasil transformasi dataset dari csv menjadi Matriks *User-Item Rating*, didapat Matriks berukuran 6660 baris x 10 kolom. Kategori-kategori barang yang direpresentasikan oleh kolom matriks tersebut antara lain *automotive, beauty, books, clothing, electronics, groceries, home & kitchen, pet supplies, sports*, dan *toys*.

C. Membuat Matriks Rating Pengguna Baru

Kemudian, program menerima input baru sejumlah kolom Matriks *User-Item Rating* yang menyimulasikan *rating* dari pengguna baru *e-commerce* tersebut pada kategori-kategori barang yang ada. Pengguna baru bisa memilih untuk memberikan *rating* pada suatu kategori barang, memberikan *rating* dianggap sebagai pengguna tersebut pernah membeli barang yang diberi *rating*, atau tidak memberikan *rating*, tidak memberikan *rating* dianggap sebagai pengguna tersebut belum pernah membeli barang yang tidak diberi *rating*. Pada program ini, tidak memberikan *rating* pada suatu kategori barang dilakukan dengan memberi masukan 's' atau "skip".

```
def getUserInput (cols):
    lenCols = len(cols)
    userInput = np.zeros((1, lenCols))
    validInput = np.zeros((1, lenCols), dtype=bool)
    for i in range(lenCols):
        while (True):
            try:
                valStr = input(f"Masukkan rating (0 - 5) terhadap item "
                               f"{cols[i]} (masukkan s atau skip untuk melewati)\n")

                if (valStr == "s" or valStr == "skip") :
                    validInput[0][i] = False
                    break

                valFloat = float(valStr)

                if (valFloat >= 0 and valFloat <= 5) :
                    validInput[0][i] = True
                    break
                else:
                    print("Masukan harus berada di rentang 0 - 5")

            except ValueError:
                print("Masukan harus berupa angka dalam rentang 0 - 5")

        if (validInput[0][i]) :
            userInput[0][i] = valFloat

    return userInput, validInput
```

Gambar 3.2. Subprogram untuk menerima input yang menyimulasikan pengguna baru *e-commerce*

D. Menentukan Batasan-batasan

Selanjutnya ditentukan batasan-batasan dari program. Batasan-batasan ini mencakup ukuran dimensi laten (k), jumlah iterasi ($iter$), dan batas toleransi yang diterima ($toleransi$) agar program tidak melakukan *loop* yang tidak bermakna.

E. Inisialisasi Matriks Acak W dan H

Langkah selanjutnya adalah menginisiasi matriks W dan H yang akan menjadi hasil faktorisasi dari Matriks *User-Item Rating*. Matriks W diinisiasi dengan matriks acak berukuran $row * k$ dan matriks H diinisiasi dengan matriks acak berukuran $k * col$.

```
iter = 100
k = 5
toleransi = 1e-9

W = np.random.rand(row, k)
H = np.random.rand(k, col)
```

Gambar 3.3. Potongan program yang menunjukkan penetapan batasan serta inisialisasi matriks W dan H

F. Terapkan Aturan Iteratif

Setelah mendapatkan matriks awal W dan H , langkah selanjutnya adalah menerapkan aturan iterasi dari *Lee* dan *Seung* pada kedua matriks tersebut sebanyak jumlah iterasi yang telah ditentukan sebelumnya. Iterasi akan dihentikan apabila nilai norma frobenius dari selisih antara Matriks *User-Item Rating* yang awal dengan matriks $W*H$ lebih kecil dari batas toleransi yang telah ditentukan sebelumnya pula. Hal ini dilakukan karena batas toleransi tersebut dianggap sudah sangat kecil sehingga program tidak perlu melakukan komputasi lebih lanjut lagi.

```
for i in range(iter) :
    W = transformW(H, W, V) # W_baru
    H = transformH(H, W, V) # H_baru

    if (frobeniusNorm(V - W @ H) < toleransi) :
        break
```

Gambar 3.4. Potongan program yang menunjukkan penerapan aturan iterasi dari *Lee* dan *Seung*

G. Ambil Hasil Rekomendasi

Setelah iterasi selesai, program akan memberikan lima rekomendasi barang yang diprediksi akan disukai oleh pengguna baru dengan memanfaatkan hasil faktorisasi matriks nonnegatif. Rekomendasi akan diberikan pada barang-barang yang belum pernah dibeli (diberi *rating*) oleh pengguna baru tersebut.

```
Rekomendasi Item Untuk User:
1. Toys : (1.2676 / 5.0)
2. Pet Supplies : (0.8547 / 5.0)
3. Home & Kitchen : (0.6984 / 5.0)
4. Electronics : (0.6968 / 5.0)
5. Groceries : (0.1403 / 5.0)
```

Gambar 3.5. Contoh hasil list rekomendasi yang diberikan

IV. HASIL IMPLEMENTASI DAN PENGUJIAN

A. Kasus Uji I

Kasus pertama yang akan diuji adalah sebagai berikut:

Kategori Barang	Rating yang diberikan
<i>Automotive</i>	s
<i>Beauty</i>	s
<i>Books</i>	s
<i>Clothing</i>	s
<i>Electronics</i>	s
<i>Groceries</i>	s
<i>Home & Kitchen</i>	s
<i>Pet Supplies</i>	s
<i>Sports</i>	s
<i>Toys</i>	s

Dari data kasus uji tersebut, didapatkan hasil sebagai berikut:

```
Rekomendasi Item Untuk User:
1. Automotive : (nan / 5.0)
2. Beauty : (nan / 5.0)
3. Books : (nan / 5.0)
4. Clothing : (nan / 5.0)
5. Electronics : (nan / 5.0)
```

Gambar 4.1. Hasil list rekomendasi untuk kasus uji pertama

Kasus uji pertama ini menunjukkan kasus ketika pengguna baru benar-benar belum pernah membeli apa pun pada *e-commerce* tersebut. Karena belum ada data yang bisa diproses, program tidak bisa

memberikan rekomendasi dengan benar sehingga hasil list rekomendasi yang diberikan hanyalah 5 kategori paling awal dengan nilai rekomendasinya sebesar nan (*not a number*).

B. Kasus Uji II

Kasus kedua yang akan diuji adalah sebagai berikut:

Kategori Barang	Rating yang diberikan
<i>Automotive</i>	s
<i>Beauty</i>	s
<i>Books</i>	5
<i>Clothing</i>	s
<i>Electronics</i>	s
<i>Groceries</i>	s
<i>Home & Kitchen</i>	s
<i>Pet Supplies</i>	s
<i>Sports</i>	s
<i>Toys</i>	s

Dari data kasus uji tersebut, didapatkan hasil sebagai berikut:

```
Rekomendasi Item Untuk User:
1. Electronics : (1.7547 / 5.0)
2. Groceries : (1.3625 / 5.0)
3. Home & Kitchen : (0.6944 / 5.0)
4. Sports : (0.4967 / 5.0)
5. Automotive : (0.0348 / 5.0)
```

Gambar 4.2. Hasil list rekomendasi untuk kasus uji kedua

Pada kasus uji kedua, sudah ada data yang bisa diolah oleh program. Walaupun hanya terdapat satu data (datanya *sparse*), program tetap bisa memberikan rekomendasi dengan normal karena metode ini tetap bisa bekerja dengan baik bahkan pada data yang *sparse* sekalipun.

C. Kasus Uji III

Kasus ketiga yang akan diuji adalah sebagai berikut:

Kategori Barang	Rating yang diberikan
<i>Automotive</i>	4.1
<i>Beauty</i>	2.3
<i>Books</i>	5
<i>Clothing</i>	0.4
<i>Electronics</i>	1.1
<i>Groceries</i>	3
<i>Home & Kitchen</i>	4.2
<i>Pet Supplies</i>	2
<i>Sports</i>	2.1
<i>Toys</i>	3.4

Dari data kasus uji tersebut, didapatkan hasil sebagai berikut:

```
Rekomendasi Item Untuk User:
```

Gambar 4.3. Hasil list rekomendasi untuk kasus uji ketiga

Pada kasus uji ini, pengguna tidak mendapatkan rekomendasi barang apa pun. Hal ini disebabkan program hanya memberi rekomendasi pada barang yang belum pernah dibeli (belum diberi *rating*) oleh pengguna tersebut. Karena pada kasus uji ini, pengguna telah memberi *rating* pada semua kategori barang, list rekomendasi yang diberikan program menjadi kosong.

D. Kasus Uji IV

Kasus keempat yang akan diuji adalah sebagai berikut:

Kategori Barang	Rating yang diberikan
<i>Automotive</i>	0
<i>Beauty</i>	s
<i>Books</i>	0
<i>Clothing</i>	0
<i>Electronics</i>	s
<i>Groceries</i>	s
<i>Home & Kitchen</i>	0
<i>Pet Supplies</i>	0
<i>Sports</i>	s
<i>Toys</i>	s

Dari data kasus uji tersebut, didapatkan hasil sebagai berikut:

```
Rekomendasi Item Untuk User:
1. Beauty : (nan / 5.0)
2. Electronics : (nan / 5.0)
3. Groceries : (nan / 5.0)
4. Sports : (nan / 5.0)
5. Toys : (nan / 5.0)
```

Gambar 4.4. Hasil list rekomendasi untuk kasus uji keempat

Hasil yang diberikan pada kasus uji ini mirip dengan kasus uji yang pertama. Hal ini disebabkan *rating* yang diberikan pengguna tersebut pada semua barang yang telah ia beli adalah nol. Nilai nol tersebut tidak memberikan informasi yang berguna pada program sehingga program tidak bisa memberikan list rekomendasi yang benar.

V. KESIMPULAN

Dari hasil-hasil pengujian yang didapatkan, bisa disimpulkan bahwa hasil rekomendasi yang diberikan menggunakan algoritma ini masih memiliki keterbatasan yang banyak sehingga tentu belum bisa dikatakan sempurna. Hal ini dapat dilihat dari beberapa kasus uji yang tidak berhasil memberikan list rekomendasi yang sesuai kepada pengguna apabila data yang diberikan memiliki kecacatan, seperti semua masukan bernilai nol dan lain sebagainya.

Walaupun demikian, hasil pengujian juga dapat menunjukkan bahwa metode ini tetap dapat menangkap

pola hubungan antara pengguna dengan barang yang dibelinya selama data yang diberikan lengkap dan cukup bermakna. Dengan kata lain, metode ini tetap bisa memberikan hasil rekomendasi yang relevan terhadap preferensi pembelian pengguna.

Terlepas dari ketidaksempurnaan pemberian rekomendasi tersebut, metode faktorisasi matriks nonnegatif ini bisa dianggap sebagai metode awal yang relatif sederhana dan mudah diimplementasikan. Untuk penelitian selanjutnya, mungkin metode ini bisa digabungkan dengan metode-metode pemberian rekomendasi yang lain sehingga lebih bisa memberikan hasil rekomendasi yang relevan sesuai kebutuhan dan keinginan peserta. Hal ini terutama karena metode ini sangat bergantung pada pemberian *rating* pada produk yang sudah pernah dibeli oleh pengguna tersebut sebelumnya.

VI. LAMPIRAN

Tautan github :

<https://github.com/MFakhryzaKi/MakalahAlgeo>

VII. UCAPAN TERIMA KASIH

Saya ingin mengucapkan terima kasih sebesar-besarnya terutama kepada Tuhan Yang Maha Esa, Allah Swt, yang telah menganugrahi saya kemampuan untuk bisa menyelesaikan makalah ini. Selanjutnya ucapan terima kasih juga ingin saya sampaikan kepada dosen saya Bapak Ir. Rila Mandala, M.Eng., Ph.D. yang telah dengan sabar mengajarkan mata kuliah Aljabar Linier dan Geometri kepada saya. Terakhir, saya juga ingin mengucapkan terima kasih kepada orang tua dan teman-teman saya yang senantiasa memberikan saya dukungan dalam bentuk apa pun.

REFERENSI

- [1] Lee, Seung. 2000. "Algorithms for Non-negative Matrix Factorization".
https://proceedings.neurips.cc/paper_files/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf
(Diakses pada 21 Desember 2025)
- [2] Munir, Rinaldi. 2025 "Review Matriks".
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-01-Review-Matriks-2025.pdf>
(Diakses pada 24 Desember 2025)
- [3] Munir, Rinaldi. 2025 "Singular Value Decomposition (SVD)".
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-21-Singular-value-decomposition-Bagian1-2025.pdf>
(Diakses pada 24 Desember 2025)
- [4] Munir, Rinaldi. 2025 "Dekomposisi LU".
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-23-Dekomposisi-LU-2025.pdf>
(Diakses pada 24 Desember 2025)
- [5] Munir, Rinaldi. 2025 "Dekomposisi QR".
<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-24-Dekomposisi-QR-2025.pdf>
(Diakses pada 24 Desember 2025)
- [6] Prayogo, Hadi, Azzahra. 2024 "Komparasi Metode Dekomposisi Matriks Menilai Kelebihan dan Kekurangan SVD, QR, dan LU dalam Aplikasi Aljabar Linear"
<https://ejournal.upi.edu/index.php/JEM/article/view/75857>

(Diakses pada 24 Desember 2025)

[7] E-commerce_dataset.

<https://www.kaggle.com/datasets/abhayayare/e-commerce-dataset>

(Diakses pada 21 Desember 2025)

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.

Bandung, 24 Desember 2025



Muhammad Fakhry Zaki
13524087