

# Penerapan Dekomposisi QR dalam Meningkatkan Akurasi Sistem Rekomendasi Berbasis *Collaborative Filtering*

Heleni Gratia Meitrina Tampubolon - 13523107<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

[113523107@std.stei.itb.ac.id](mailto:113523107@std.stei.itb.ac.id), [helenigratia@gmail.com](mailto:helenigratia@gmail.com)

**Abstrak**— Sistem rekomendasi adalah teknologi berbasis algoritma yang bertujuan memberikan rekomendasi yang relevan kepada pengguna dengan mempertimbangkan preferensi, perilaku, atau data historis mereka. Teknologi ini telah diterapkan secara luas di berbagai platform digital, seperti *e-commerce*, layanan streaming, dan media sosial. Salah satu pendekatan yang digunakan dalam pengembangan sistem rekomendasi adalah *collaborative filtering*. Dalam pendekatan ini, algoritma dekomposisi QR dan *Alternating Least Square (ALS)* diterapkan untuk meningkatkan efisiensi dan akurasi. Kombinasi kedua algoritma tersebut mampu mengoptimalkan pengolahan dataset yang berskala besar dan bersifat jarang (*sparse*).

**Kata Kunci**—Dekomposisi QR, ALS, *Collaborative Filtering*, Sistem Rekomendasi.

## I. PENDAHULUAN

Sistem rekomendasi adalah sistem yang dirancang untuk memberikan saran item yang relevan kepada pengguna, seperti produk, musik, atau acara televisi. Dengan memanfaatkan machine learning dan teknik penyaringan data, sistem ini mampu memprediksi kebutuhan dan preferensi pengguna secara akurat. Keberadaan sistem rekomendasi menjadi sangat penting di era digital, di mana pengguna dihadapkan pada jumlah informasi yang melimpah. Dengan memberikan rekomendasi yang relevan dan personal, sistem ini membantu mengatasi masalah *information overload* sekaligus meningkatkan kenyamanan pengguna dalam menemukan konten yang sesuai.

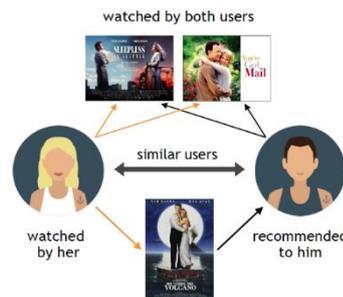
Rekomendasi yang diberikan oleh sistem ini didasarkan pada berbagai sumber data, termasuk riwayat pembelian, aktivitas pencarian, informasi demografi, serta pola interaksi seperti jumlah klik, suka, atau pembelian sebelumnya. Sistem rekomendasi dilatih untuk memahami preferensi pengguna dengan menganalisis pola keputusan yang mereka buat sebelumnya dan perilaku mereka saat berinteraksi dengan produk atau layanan tertentu. Hal ini menjadikan sistem mampu menawarkan saran yang relevan secara kontekstual dan personal.

Terdapat beberapa pendekatan untuk membangun sistem rekomendasi, salah satunya adalah *collaborative filtering (CF)*. *Collaborative filtering* bekerja dengan memberikan rekomendasi (proses penyaringan) berdasarkan data preferensi dari berbagai pengguna (proses kolaborasi). Prinsip utama dari metode ini adalah kesamaan pola: jika sekelompok pengguna

memiliki preferensi yang serupa dalam memilih suatu item, maka ada kemungkinan besar bahwa item lain yang disukai kelompok tersebut juga akan disukai oleh pengguna lainnya.

Metode *collaborative filtering* dapat memanfaatkan dua jenis data interaksi::

1. *Explicit feedback*, seperti ulasan atau penilaian (rating) yang diberikan secara langsung oleh pengguna.
2. *Implicit feedback*, seperti data klik, riwayat pembelian, waktu menonton, atau perilaku lainnya yang secara tidak langsung menunjukkan preferensi pengguna.



Gambar 1.1 Sistem Rekomendasi Berbasis Collaborative Filtering

Sumber: <https://www.nvidia.com/en-us/glossary/recommendation-system/>

Penerapan metode *collaborative filtering* telah banyak diadopsi dalam berbagai aplikasi selama beberapa tahun terakhir. Salah satu contohnya adalah platform *e-commerce* seperti Amazon, yang menggunakan metode ini untuk memberikan rekomendasi produk berdasarkan preferensi pengguna dan pola pembelian sebelumnya. Di sisi lain, layanan streaming seperti Netflix telah menerapkan *hybrid recommender system*, yang mengombinasikan *collaborative filtering* dan *content-based filtering*, sejak tahun 2009. Pendekatan ini memungkinkan Netflix untuk memberikan rekomendasi film dan serial yang lebih akurat, sesuai dengan preferensi pengguna sekaligus memperhitungkan karakteristik konten.

Namun, implementasi metode *collaborative filtering* dalam sistem rekomendasi tidak terlepas dari tantangan. Salah satu tantangan utama adalah ukuran dataset yang sangat besar, terutama pada platform yang memiliki jutaan pengguna dan item. Tantangan lainnya adalah matriks *utility* yang terbentuk cenderung sangat *sparse* karena sebagian besar pengguna tidak

memberikan rating pada semua item. Sparsity ini dapat menyebabkan algoritma rekomendasi menjadi kurang efisien atau akurat.

Untuk mengatasi masalah tersebut, teknik *Matrix Factorization* menjadi solusi yang sering digunakan. Dengan memfaktorkan matriks *utility* ke dalam dua matriks lebih kecil, metode ini dapat menangkap pola laten yang merepresentasikan hubungan antara pengguna dan item. Salah satu pendekatan populer dalam *Matrix Factorization* adalah *Singular Value Decomposition* (SVD), yang telah banyak dikaji dalam berbagai penelitian dan jurnal ilmiah.

Namun, dalam makalah ini, metode *Matrix Factorization* akan diselesaikan menggunakan pendekatan alternatif, yaitu dekomposisi QR. Penggunaan dekomposisi QR menawarkan keuntungan berupa stabilitas numerik dan efisiensi untuk penyelesaian sistem linier, yang menjadi dasar dalam memperbarui parameter dalam algoritma *Alternating Least Squares* (ALS). Dengan metode ini, diharapkan dapat memberikan perspektif baru dalam pemanfaatan dekomposisi matriks untuk mengatasi tantangan pada sistem rekomendasi berbasis *collaborative filtering*.

## II. LANDASAN TEORI

### A. Matriks

#### a. Defenisi Matriks

Matriks merupakan representasi data dalam bentuk susunan elemen berupa angka, simbol, atau karakter yang tersusun secara sistematis dalam baris dan kolom. Elemen-elemen dalam matriks diidentifikasi berdasarkan posisi spesifiknya yang ditentukan oleh indeks baris dan kolom. Representasi matriks ini memungkinkan pengolahan data secara terstruktur. Matriks biasanya ditulis menggunakan tanda kurung siku atau kurung biasa, dan setiap elemen dalam matriks merepresentasikan nilai tertentu yang berkaitan dengan konteks penggunaannya.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} \\ a_{21} & a_{22} & \dots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} \end{bmatrix}$$

→ baris 1  
→ baris 2  
→ baris i

↓ kolom 1  
↓ kolom 2  
↓ kolom j

Gambar 2.1. Matriks

Sumber: <https://akupintar.id/info-pintar/-/blogs/matriks-pengertian-operasi-determinan-invers-dan-contoh-soal>

#### b. Istilah-Istilah Matriks

Sebuah matriks memiliki karakteristik yang dapat dijelaskan melalui dua aspek utama:

1. Ukuran (dimensi), Ukuran matriks ditentukan oleh jumlah baris ( $m$ ) dan jumlah kolom ( $n$ ), yang kemudian ditulis sebagai matriks berukuran  $m \times n$ . Matriks dengan ukuran tertentu merepresentasikan susunan elemen di dalam baris dan kolom yang sesuai. Sebagai contoh, matriks berukuran  $3 \times 2$  berarti memiliki 3 baris dan 2 kolom.
2. Elemen, setiap elemen dalam matriks diwakili oleh

notasi  $a_{ij}$ , di mana  $i$  menunjukkan indeks baris dan  $j$  menunjukkan indeks kolom. Elemen ini merupakan nilai spesifik yang berada di persimpangan baris ke- $i$  dan kolom ke- $j$ .

Dalam materi matriks, terdapat beberapa istilah penting yang sering digunakan untuk menggambarkan sifat dan karakteristik matriks:

1. Ordo, merupakan ukuran matriks yang didapatkan dari perkalian banyaknya baris dengan banyaknya kolom pada matriks. Contoh: matriks A memiliki  $m$  baris dan  $n$  kolom, maka matriks A berordo  $m \times n$ . Jika matriks merupakan matriks persegi dengan  $m$  baris, maka ordo matriks dapat langsung disebut ordo  $m$ .
2. Determinan, merupakan nilai skalar yang dihitung dari matriks persegi. Determinan matriks A ditulis dengan  $\det(A)$  atau  $|A|$ .

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \det(A) = ad - bc$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$\det(A) = a_{11} \det \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} - a_{12} \det \begin{pmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{pmatrix} + a_{13} \det \begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}$$

$$= a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$$

Gambar 2.2. Determinan Matriks 2x2 dan 3x3

Sumber: <https://mxncalc.com/determinant-calculator>

3. Trace matriks, merupakan jumlah elemen-elemen yang berada pada diagonal utama dari sebuah matriks persegi.

#### c. Jenis-Jenis Matriks

Matriks terbagi menjadi beberapa jenis, diantaranya:

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \rightarrow \text{matriks nol} \quad F = \begin{bmatrix} 1 & 7 & 4 \\ 0 & 5 & 9 \\ 0 & 0 & 7 \end{bmatrix} \rightarrow \text{matriks segitiga atas}$$

$$B = [5 \quad 3] \rightarrow \text{matriks baris}$$

$$C = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \rightarrow \text{matriks kolom} \quad G = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \\ 7 & 6 & 9 \end{bmatrix} \rightarrow \text{matriks segitiga bawah}$$

$$D = \begin{bmatrix} 3 & 2 \\ 7 & 0 \end{bmatrix} \rightarrow \text{matriks persegi} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{matriks identitas}$$

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{bmatrix} \rightarrow \text{matriks diagonal}$$

Gambar 2.3. Jenis-Jenis Matriks

Sumber: koleksi pribadi

1. Matriks nol, matriks yang seluruh elemennya adalah bilangan nol.
2. Matriks baris, matriks yang hanya memiliki satu baris, berorde  $1 \times j$ .
3. Matriks kolom, matriks yang hanya memiliki satu kolom, berorde  $i \times 1$ .
4. Matriks persegi, matriks yang banyaknya baris sama dengan banyaknya kolom, berorde  $i \times i$ .
5. Matriks diagonal, matriks persegi yang semua elemennya nol, kecuali pada diagonal utamanya.
6. Matriks segitiga atas, matriks persegi yang semua elemen

di bawah diagonal utama adalah nol.

7. Matriks segitiga bawah, matriks persegi yang semua elemen di atas diagonal utamanya adalah nol.
8. Matriks identitas, matriks persegi yang elemen pada diagonal utamanya adalah satu, sedangkan elemen lainnya adalah nol.
9. Matriks transpose, matriks yang diperoleh dengan menukar baris menjadi kolom.
10. Matriks orthogonal, matriks yang setiap kolomnya adalah vector sedemikian sehingga hasil kali titik setiap vector dengan vector lainnya adalah 0.
11. Matriks ortonormal adalah matriks yang kolom-kolomnya membentuk himpunan orthogonal dan setiap vector merupakan vector satuan.
12. Matriks singular, matriks persegi yang determinannya bernilai nol.

#### d. Operasi Matriks

Operasi matriks adalah tindakan aljabar yang dilakukan pada dua atau lebih matriks untuk menghasilkan matriks baru atau nilai tertentu. Operasi ini dilakukan berdasarkan aturan-aturan khusus yang berlaku untuk matriks, termasuk ketentuan terkait ukuran (ordo) matriks. Operasi-operasi ini meliputi penjumlahan, pengurangan, perkalian, transpos, invers, dan operasi lainnya. Setiap operasi harus mematuhi sifat-sifat matriks dan keterbatasan dimensi untuk memastikan hasil yang valid dan konsisten.

##### 1. Penjumlahan Matriks

Dua matriks dapat dijumlahkan jika dan hanya jika ukurannya sama ( $m \times n$ ). Untuk menjumlahkan dua matriks, dilakukan penjumlahan untuk setiap elemen matriks yang bersesuaian lokasi atau urutannya.

$$\begin{bmatrix} 1 & 4 \\ 3 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 7 \\ 9 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 11 \\ 12 & 6 \end{bmatrix}$$

$$A + B = C$$

Gambar 2.4. Penjumlahan Matriks

Sumber: koleksi pribadi

Dalam operasi penjumlahan matriks, terdapat beberapa sifat yang memenuhi untuk matriks A, B, dan C dengan orde yang sama.

- a. Hukum Komutatif:  $A + B = B + A$
- b. Hukum Asosiatif:  $(A + B) + C = A + (B + C)$
- c. Identitas Matriks:  $A + O = O + A = A$ , dengan O merupakan matriks nol.
- d. Invers Aditif:  $A + (-A) = O = (-A) + A$ , dengan (-A) diperoleh dengan mengubah tanda setiap elemen A, yang merupakan invers aditif dari matriks.

##### 2. Pengurangan Matriks

Dua matriks dapat dikurangkan jika dan hanya jika ukurannya sama ( $m \times n$ ). Untuk mengurangkan dua matriks, dilakukan pengurangan untuk setiap elemen matriks yang bersesuaian lokasi atau urutannya.

$$\begin{bmatrix} 3 & 2 \\ 11 & 7 \end{bmatrix} - \begin{bmatrix} 2 & 6 \\ 5 & 3 \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 6 & 4 \end{bmatrix}$$

$$A - B = C$$

Gambar 2.5. Pengurangan Matriks

Sumber: koleksi pribadi

##### 3. Perkalian Skalar Matriks

Untuk setiap matriks, jika dilakukan perkalian dengan scalar apapun (misalkan k) maka scalar tersebut dikalikan dengan setiap elemen matriks.

$$kA = \begin{bmatrix} k \cdot 2 & k \cdot 6 \\ k \cdot 5 & k \cdot 3 \end{bmatrix}$$

Gambar 2.6. Perkalian Skalar Matriks

Sumber: koleksi pribadi

Dalam operasi perkalian skalar, matriks apa pun, A serta  $\alpha$  dan  $\beta$  adalah skalar, maka:

- a.  $(\alpha + \beta)A = \alpha A + \beta A$
- b.  $\alpha(\beta A) = (\alpha\beta)A = \beta(\alpha A)$
- c.  $(-\alpha A) = -(\alpha A) = \alpha(-A)$

##### 4. Perkalian Matriks

Perkalian matriks hanya dapat dilakukan jika jumlah kolom pada matriks pertama sama dengan jumlah baris pada matriks kedua, yaitu matriks  $A_{m \times n}$  dan matriks  $A_{n \times o}$ . Perkalian matriks juga tidak bersifat komutatif ( $A \times B \neq B \times A$ ).

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \times \begin{bmatrix} p & q & r \\ s & t & u \end{bmatrix} = \begin{bmatrix} ap + bs & aq + bt & ar + bu \\ cp + ds & cq + dt & cr + du \\ ep + fs & eq + ft & er + fu \end{bmatrix}$$

Gambar 2.6. Perkalian Matriks

Sumber: koleksi pribadi

##### 5. Transpose Matriks

Operasi transpose matriks merupakan operasi yang menukar kolom matriks menjadi baris matriks dan sebaliknya secara beraturan sesuai lokasi elemen matriks. Matriks hasil transpose biasanya ditulis menggunakan notasi  $A^T$ .

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Gambar 2.7. Transpose Matriks

Sumber: koleksi pribadi

##### 6. Invers Matriks

Untuk setiap matriks A, invers hanya dapat ditemukan jika A adalah matriks persegi dan determinannya sama dengan 1. Metode dalam mencari invers matriks dapat berupa metode adjoin, eliminasi gauss-jordan, dan lainnya. Matriks ini direpresentasikan sebagai  $(A)^{-1}$  dan jika dilakukan perkalian dengan matriks A maka memperoleh matriks identitas (I).

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ dengan } ad - bc \neq 0$$

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{\det A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Gambar 2.8. Invers Matriks

Sumber: koleksi pribadi

##### 7. Norm Vektor

Norm vector adalah ukuran panjang atau magnitude vektor. Norm didefinisikan sebagai:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2} \quad (1)$$

dengan  $\mathbf{v} = [v_1, v_2, \dots, v_n]$  adalah vektor dan  $\|\mathbf{v}\|$  adalah norm-nya.

### 8. Normalisasi Vektor

Normalisasi dilakukan dengan membagi setiap elemen vektor dengan norm-nya, sehingga diperoleh vektor satuan dengan panjang 1.

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|} = \left[ \frac{v_1}{\|\mathbf{v}\|}, \frac{v_2}{\|\mathbf{v}\|}, \dots, \frac{v_n}{\|\mathbf{v}\|} \right] \quad (2)$$

### B. Dekomposisi QR

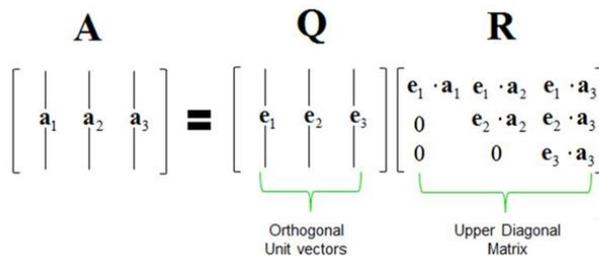
Dekomposisi QR adalah pemfaktoran matriks berukuran  $m \times n$  menjadi hasil kali matriks orthogonal dan matriks segitiga.

$$A = QR, \quad (3)$$

$Q = \text{matriks ortonormal}, \quad R = \text{matriks segitiga atas}$

Salah satu metode untuk menghitung dekomposisi QR adalah metode Gram-Schmidt.

Jika ditinjau sebuah matriks  $A$  berukuran  $m \times n$ . Setiap kolom pada matriks  $A$  dipandang sebagai vector:  $A = [a_1 \ a_2 \ a_n]$ . Bentuk hasil faktorisasi QR adalah:



Gambar 2.9. Invers Matriks

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/Algeo-23b-Dekomposisi-QR-2024.pdf>

Elemen  $e$  dan  $a$  diperoleh dengan pola sebagai berikut:

$$\begin{aligned} u_1 &= a_1, & e_1 &= \frac{u_1}{\|u_1\|}, \\ u_2 &= a_2 - (a_2 \cdot e_1)e_1, & e_2 &= \frac{u_2}{\|u_2\|}, \\ u_{k+1} &= a_{k+1} - (a_{k+1} \cdot e_1)e_1 - \dots - (a_{k+1} \cdot e_k)e_k, & e_{k+1} &= \frac{u_{k+1}}{\|u_{k+1}\|}. \end{aligned}$$

Gambar 2.10. Invers Matriks

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/Algeo-23b-Dekomposisi-QR-2024.pdf>

### C. Sistem Rekomendasi

Sistem rekomendasi adalah jenis sistem penyaringan informasi yang dirancang untuk memperkirakan preferensi pengguna dan memberikan saran yang relevan berdasarkan data yang ada. Keberadaan sistem rekomendasi sangat penting dalam mengatasi masalah kelebihan informasi yang sering dihadapi oleh pengguna, dengan menyajikan rekomendasi konten atau layanan yang dipersonalisasi sesuai dengan kebutuhan atau preferensi masing-masing. Sistem ini telah diterapkan secara luas dalam berbagai platform digital, termasuk e-commerce,

jejaring sosial, dan layanan streaming, seperti yang terlihat pada Amazon dan Netflix.

Sistem rekomendasi bekerja dengan menganalisis data pengguna, seperti histori penelusuran, ulasan, pembelian, dan interaksi lainnya, untuk mengidentifikasi pola preferensi yang dapat digunakan untuk memberikan rekomendasi yang lebih akurat. Misalnya, Amazon menggunakan sistem rekomendasi untuk menyarankan produk kepada pengguna berdasarkan riwayat penelusuran dan pembelian mereka.

Dalam penerapannya, terdapat beberapa metode yang digunakan dalam sistem rekomendasi, antara lain:

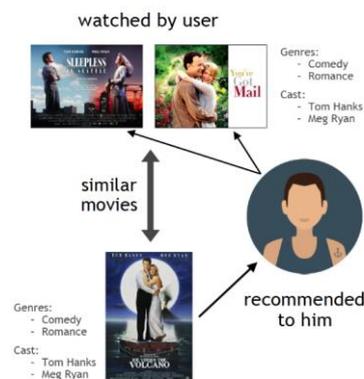
#### 1. Collaborative Filtering

Metode *collaborative filtering* (CF) adalah salah satu pendekatan utama dalam sistem rekomendasi yang bertujuan untuk merekomendasikan item atau produk berdasarkan kesamaan preferensi antar pengguna. Pada dasarnya, metode ini memanfaatkan data interaksi pengguna (seperti rating, klik, atau pembelian) untuk mencari pola-pola kesamaan antar pengguna atau item. Terdapat dua jenis utama dalam metode *collaborative filtering*, yaitu berbasis pengguna (*user-based*) dan berbasis item (*item-based*).

- Collaborative Filtering Berbasis Pengguna:** Dalam metode ini, sistem rekomendasi mencari pengguna lain yang memiliki pola preferensi yang mirip dengan pengguna target. Setelah menemukan pengguna yang memiliki kesamaan preferensi, sistem akan merekomendasikan item yang telah disukai atau dinilai tinggi oleh pengguna-pengguna tersebut, dengan asumsi bahwa pengguna target mungkin akan menyukai item yang sama.
- Collaborative Filtering Berbasis Item:** Pada metode berbasis item, sistem akan mengevaluasi kesamaan antar item berdasarkan interaksi pengguna dengan item tersebut. Jika dua item sering dipilih atau diberi rating yang serupa oleh pengguna yang sama, maka kedua item ini dianggap memiliki kesamaan. Dengan demikian, ketika pengguna memberikan rating tinggi pada suatu item, sistem dapat merekomendasikan item lain yang serupa.

#### 2. Content-Based Filtering

Metode *content-based filtering* merekomendasikan item atau produk berdasarkan karakteristik atau fitur sebelumnya yang telah disukai pengguna. Sistem ini menganalisis data terkait item, seperti deskripsi, genre, atau aktor.



Gambar 2.11. Sistem Rekomendasi Content-Based Filtering

Sumber: <https://www.nvidia.com/en-us/glossary/recommendation-system/>

### 3. Hybrid System

Sistem rekomendasi yang menggabungkan metode collaborative dan content-based. Penggabungan metode ini diharapkan dapat memanfaatkan tiap kelebihan yang dimiliki oleh masing-masing metode dan menghasilkan rekomendasi yang lebih akurat.

#### D. Alternating Least Square (ALS)

Sistem rekomendasi merupakan algoritma untuk melakukan optimisasi dalam penyelesaian masalah faktorisasi, salah satu aplikasi yang sering digunakan dalam Collaborative Filtering (CF). Tujuan dari ALS adalah menemukan dua matriks factor yang mendekati matriks yang lebih besar dan biasanya berupa *sparse* (jarang). ALS ini sangat efektif untuk menangani dataset, seperti matriks interaksi antara pengguna dan item dalam rekomendasi film, musik, atau produk.

user/item	4	5	10	14	15	18	19	20	21	24	...	2003199	2003204	2003206	2003208	2003210	2003215	2003219	2003222	2003224
1	4.0	NaN	5.0	4.0	NaN	NaN	NaN	5.0	NaN	5.0	NaN	NaN	NaN	4.5	NaN	4.0	4.0	NaN	NaN	NaN
2	4.0	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN							
3	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	NaN	4.0	NaN						
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN											
5	2.0	NaN	NaN	NaN	NaN	5.0	NaN	NaN	NaN	NaN										
6	4.5	NaN	NaN	NaN	NaN	NaN	NaN	3.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	5.0	NaN	NaN	NaN											
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN											
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN											
10	4.0	NaN	NaN	NaN	NaN	NaN	NaN	3.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	3.5	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	NaN	NaN	NaN
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN											
13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN											
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.0											
15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN											

Gambar 2.12. Data Sparse

Sumber: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1>

Dalam penerapan di Collaborative Filtering (CF), konsep ALS ialah memfaktorkan sebuah matriks yang berukuran besar menjadi dua matriks yang lebih kecil, contohnya matriks pengguna dan matriks item.

1. Matriks pengguna (user matrices) (U), merupakan matriks berukuran  $m \times k$  berisi preferensi pengguna (m) terhadap berbagai factor laten (k), seperti genre atau jenis film.
2. Matriks item (item matrices) (V), merupakan matriks berukuran  $n \times k$  berisi karakteristik item (n) dalam ruang faktor laten yang sama.

Tujuan dari ALS ialah memperkirakan matriks yang hilang dengan meminimalkan kesalahan kuadrat antara matriks yang sebenarnya dan perkiraan hasil perkalian  $U \times V^T$ .

Langkah-langkah dalam penerapan *Alternating Least Square* (ALS):

1. Inisialisasi, matriks pengguna (U) dan matriks item (V) diinisialisasi dengan nilai acak.
2. *Alternating Minimization*, terdiri dari dua langkah. Langkah pertama, nilai matriks V dianggap konstan dan menghitung U dengan meminimalkan loss function:

$$\min_{X,Y} \sum_{r_{ui} \text{ observed}} (r_{ui} - x_u^T y_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

Gambar 2.13. Loss Function

Sumber:

<https://web.stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf>

Rumus tersebut terdiri dari dua bagian, bagian pertama merupakan nilai error antara nilai asli matriks R dan perkiraan dari U dan V, sedangkan bagian kedua adalah *regularization term* ( $\lambda$ ) untuk mencegah *overfitting*. Kemudian, dilakukan langkah kedua dengan menghitung nilai V dengan meminimalkan *loss function*.

3. Iterasi, langkah kedua diulang hingga konvergensi tercapai, yang berarti perubahan dalam matriks pengguna dan item menjadi sangat kecil atau sudah mencapai jumlah iterasi yang ditentukan. Setiap iterasi bertujuan untuk semakin mendekatkan hasil perkalian matriks pengguna dan item terhadap matriks yang diinginkan.

## III. IMPLEMENTASI DAN PEMBAHASAN

Penelitian ini mengusulkan pendekatan untuk meningkatkan akurasi sistem rekomendasi berbasis *Collaborative Filtering* (CF) melalui kombinasi metode dekomposisi QR dan Alternating Least Square (ALS). Dekomposisi QR digunakan untuk menyelesaikan sistem persamaan linear dengan lebih stabil dan presisi, sehingga menghasilkan solusi yang lebih akurat dan tahan terhadap kesalahan kecil dalam perhitungan. Sementara itu, ALS menjadi kerangka utama dalam proses pemfaktoran matriks interaksi pengguna-item untuk mengidentifikasi pola laten. Kombinasi kedua metode ini diharapkan mampu mengatasi tantangan data yang *sparse* (jarang) sekaligus meningkatkan kualitas prediksi dengan cara yang lebih efisien.

Dekomposisi QR diimplementasikan sebagai alternatif dari dekomposisi matriks Singular Value Decomposition (SVD), yang telah banyak digunakan dalam sistem rekomendasi. Metode ini membagi matriks menjadi dua komponen utama:

1. Matriks orthogonal (Q), membantu menjaga kestabilan perhitungan.
2. Matriks segitiga atas (R), mempermudah penyelesaian sistem persamaan linear.

Dalam konteks *Collaborative Filtering*, dekomposisi QR diterapkan pada matriks fitur pengguna atau item yang dihasilkan selama proses ALS.

Alternating Least Square (ALS) adalah algoritma iteratif yang membagi proses faktorisasi matriks menjadi dua tahap utama:

1. Memperbarui matriks fitur pengguna (U), dengan asumsi bahwa matriks fitur item (V) tetap.
2. Memperbarui matriks fitur item (V), dengan asumsi bahwa matriks fitur pengguna (U) tetap.

Proses ini diulangi hingga fungsi kerugian (*loss function*) mencapai tingkat konvergensi, yaitu ketika perbedaan antar iterasi menjadi sangat kecil.

Kombinasi dekomposisi QR dan ALS memberikan keunggulan dalam hal efisiensi, akurasi, dan stabilitas perhitungan. Penerapan ini menawarkan pendekatan baru yang mampu menggantikan SVD dalam mengatasi masalah matriks sparse, serta meningkatkan kinerja sistem rekomendasi. Algoritma ini dari program dapat diakses melalui Kaggle (<https://www.kaggle.com/code/helenigratia/book-recomendation-als-qr-method>)

### A. Dataset dan Matriks Interaksi

Dataset yang digunakan dalam penelitian ini berupa matriks

interaksi pengguna-item, di mana nilai dalam matriks merepresentasikan preferensi pengguna terhadap item tertentu, seperti *rating* film, buku, atau musik. Karena *rating* yang diberikan pengguna cenderung jarang (*sparse*), sebagian besar elemen dalam matriks ini bernilai nol. Untuk mengatasi hal ini, matriks interaksi difaktorkan menjadi dua matriks laten dengan dimensi yang lebih kecil, sehingga lebih mudah dianalisis dan diproses.

Dimensi laten (*k*) adalah jumlah fitur yang digunakan untuk merepresentasikan hubungan antara pengguna dan item dalam bentuk yang lebih sederhana. Pemilihan dimensi laten memiliki beberapa tujuan utama:

1. Menyederhanakan data, mengurangi dimensi dari matriks pengguna-item asli yang biasanya sangat besar menjadi representasi yang lebih ringkas.
2. Menangkap pola tersembunyi, membantu mengidentifikasi faktor-faktor laten yang memengaruhi preferensi pengguna, seperti penulis, popularitas, atau genre dalam konteks sistem rekomendasi buku.
3. Meningkatkan generalisasi, fokus pada faktor laten membantu menghindari *overfitting* terhadap data *sparse* dan menghasilkan prediksi yang lebih akurat untuk item yang belum pernah diberi *rating* oleh pengguna.

Namun, pemilihan nilai *k* harus dilakukan dengan hati-hati. Nilai *k* yang terlalu kecil dapat mengabaikan detail penting sehingga mengurangi akurasi model, sedangkan nilai *k* yang terlalu besar dapat menyebabkan *overfitting* dengan hasil yang terlalu spesifik. Dalam program ini, dimensi laten yang dipilih adalah *k*=2, untuk menjaga keseimbangan antara kesederhanaan dan akurasi.

Dataset yang digunakan dapat diperoleh dari platform penyedia data seperti Kaggle. Dataset yang diambil biasanya berisi informasi tentang pengguna, item, dan *rating* yang menghubungkan keduanya. Namun, data yang tersedia sering kali tidak dalam format matriks interaksi pengguna-item. Oleh karena itu, langkah pra-proses diperlukan untuk mengonversi data mentah menjadi matriks *R*, yang berisi *rating* pengguna terhadap item. Jika terdapat elemen kosong (*missing values*) dalam matriks ini, nilai tersebut akan diisi dengan angka nol, sehingga matriks dapat digunakan untuk proses pemfaktoran.

Tabel 3.1. Import Library dan Load Data pada Program

```
import numpy as np
import pandas as pd # data processing
from numpy.linalg import qr, solve
books = pd.read_csv('../input/book-recommendation-dataset/Books.csv')
users = pd.read_csv('../input/book-recommendation-dataset/Users.csv')
ratings = pd.read_csv('../input/book-recommendation-dataset/Ratings.csv')
```

Dataset yang digunakan dalam penelitian ini berasal dari Kaggle, khususnya dataset yang memuat *rating* pengguna terhadap buku. Untuk meningkatkan kualitas data dan akurasi analisis, dilakukan proses penyaringan (*filtering*) berdasarkan dua kriteria utama:

1. Judul buku, bahwa hanya buku yang telah diberi *rating* oleh  $\geq 40$  pengguna yang akan dimasukkan ke dalam analisis. Hal ini bertujuan untuk memastikan bahwa data memiliki cukup banyak interaksi pengguna

sehingga hasil prediksi menjadi lebih relevan dan dapat diandalkan.

2. Pengguna, hanya pengguna yang memberikan  $\geq 200$  *rating* terhadap berbagai buku yang akan disertakan dalam dataset. Kriteria ini membantu memfokuskan analisis pada pengguna yang memiliki riwayat interaksi yang cukup untuk menemukan pola preferensi yang jelas.

Setelah dilakukan penyaringan, elemen-elemen matriks yang tidak memiliki *rating* dari pengguna akan diisi dengan nilai nol. Pendekatan ini memungkinkan terbentuknya matriks interaksi pengguna-item yang siap digunakan untuk pemfaktoran dalam metode Alternating Least Squares (ALS).

Tabel 3.2. Pembentukan Matriks pada Program

```
import numpy as np
import pandas as pd # proses data
from numpy.linalg import qr, solve
books = pd.read_csv('../input/book-recommendation-dataset/Books.csv')
users = pd.read_csv('../input/book-recommendation-dataset/Users.csv')
ratings = pd.read_csv('../input/book-recommendation-dataset/Ratings.csv')

#menampilkan sebagian isi data beserta header
books.head()
ratings.head()
users.head()

ratings_with_name = ratings.merge(books,on='ISBN')
#Menggabungkan dua DataFrame(ratings dan books),berdasarkan ISBN.

# menghitung jumlah rating yang diberikan oleh setiap pengguna + memfilter pengguna yang memberikan >= 200 rating
x = ratings_with_name.groupby('User-ID').count()['Book-Rating'] >= 200

# Memfilter boolean pada x sehingga berisi yang nilai true saja.
wellread_users = x[x].index

filtered_rating = ratings_with_name[ratings_with_name['User-ID'].isin(wellread_users)]

#Mengelompokkan data yang sudah difilter berdasarkan judul buku yang telah diberi rating>=40 pengguna
y = filtered_rating.groupby('Book-Title').count()['Book-Rating']>=40

# Filter boolean pada y
famous_books = y[y].index

# data yang sudah di filter berdasarkan ketentuan
final_ratings = filtered_rating[filtered_rating['Book-Title'].isin(famous_books)]

# membentuk matriks dengan baris berupa User-ID dan kolom adalah judul buku
pt = final_ratings.pivot_table(index='Book-Title',columns='User-ID',values='Book-Rating')
pt.fillna(0,inplace=True)
R_panda = pt.T

# matriks R (pengguna-item)
R = R_panda.to_numpy()
```

ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M	Image-URL-L
0191513448	Classical Mechanics	Richard Fitzpatrick	2002	Oxford University Press	<a href="http://images.amazon.com/images/P/0191513448.L">http://images.amazon.com/images/P/0191513448.L</a>	<a href="http://images.amazon.com/images/P/0191513448.M">http://images.amazon.com/images/P/0191513448.M</a>	<a href="http://images.amazon.com/images/P/0191513448.L">http://images.amazon.com/images/P/0191513448.L</a>
002005019	Clara Callan	Richard Bruce Wright	2001	HarperPerennial-Canada	<a href="http://images.amazon.com/images/P/002005019.L">http://images.amazon.com/images/P/002005019.L</a>	<a href="http://images.amazon.com/images/P/002005019.M">http://images.amazon.com/images/P/002005019.M</a>	<a href="http://images.amazon.com/images/P/002005019.L">http://images.amazon.com/images/P/002005019.L</a>
006071329	Decision in Harmony	Carlo D'Este	1991	HarperPerennial	<a href="http://images.amazon.com/images/P/006071329.L">http://images.amazon.com/images/P/006071329.L</a>	<a href="http://images.amazon.com/images/P/006071329.M">http://images.amazon.com/images/P/006071329.M</a>	<a href="http://images.amazon.com/images/P/006071329.L">http://images.amazon.com/images/P/006071329.L</a>
037451760	His The Story of the Great Italian Renaissance...	Gioacchino Di Stefano	1989	Farrar Straus Giroux	<a href="http://images.amazon.com/images/P/037451760.L">http://images.amazon.com/images/P/037451760.L</a>	<a href="http://images.amazon.com/images/P/037451760.M">http://images.amazon.com/images/P/037451760.M</a>	<a href="http://images.amazon.com/images/P/037451760.L">http://images.amazon.com/images/P/037451760.L</a>
0393042128	The Mammals of Illinois	E. I. W. Butler	1989	W. W. Norton & Company	<a href="http://images.amazon.com/images/P/0393042128.L">http://images.amazon.com/images/P/0393042128.L</a>	<a href="http://images.amazon.com/images/P/0393042128.M">http://images.amazon.com/images/P/0393042128.M</a>	<a href="http://images.amazon.com/images/P/0393042128.L">http://images.amazon.com/images/P/0393042128.L</a>

Gambar 3.1. Isi Data Books  
Sumber: Koleksi Pribadi

User-ID	ISBN	Book-Rating
0	276725 034545104X	0
1	276726 0155061224	5
2	276727 0446520802	0
3	276729 052165615X	3
4	276729 0521795028	6

Gambar 3.2. Isi Data Ratings  
Sumber: Koleksi Pribadi

User-ID	Location	Age
0	1 nyc, new york, usa	NaN
1	2 stockton, california, usa	18.0
2	3 moscow, yukon territory, russia	NaN
3	4 porto, v.n.gaia, portugal	17.0
4	5 farnborough, hants, united kingdom	NaN

Gambar 3.3. Isi Data Users  
Sumber: Koleksi Pribadi

Proses yang dilakukan pada dataset akan memperoleh matriks R yang berisi matriks User-ID terhadap Book-Title:

Book-Title	1984	1st to 4th	2010	204	24	2nd	4	A 2nd	A Beautiful	Without	Wives	Word Freak	Zen and the	101
User-ID	254	2276	2766	2977	3363	27408	27970	27427	27789	27818	0.0	0.0	0.0	0.0

Gambar 3.4. Matriks R pada Dataset  
Sumber: Koleksi Pribadi

### B. Alternating Least Square (ALS)

Matriks R merepresentasikan hubungan antara pengguna dan rating yang diberikan terhadap buku. Dalam konteks metode Alternating Least Squares (ALS), matriks ini difaktorkan menjadi dua matriks yang lebih kecil, yaitu:

1. Matriks Preferensi Pengguna (P)  
Matriks ini memiliki dimensi  $num\_users \times k$ , di mana  $num\_users$  adalah jumlah pengguna, dan  $k$  adalah jumlah faktor laten. Matriks P menggambarkan preferensi masing-masing pengguna terhadap berbagai faktor laten, seperti genre, penulis, atau karakteristik lainnya yang tersembunyi dalam data.
2. Matriks Fitur Item (Q)  
Matriks ini memiliki dimensi  $num\_items \times k$ , di mana  $num\_items$  adalah jumlah buku, dan  $k$  adalah jumlah

faktor laten. Matriks Q menggambarkan karakteristik atau atribut laten yang dimiliki oleh masing-masing buku dalam ruang faktor yang sama.

Tujuan utama dari pemfaktoran ini adalah untuk memprediksi nilai rating yang belum diketahui (elemen kosong dalam RRR). Prediksi ini dilakukan dengan memperkirakan RRR sebagai hasil perkalian dua matriks tersebut:

$$R \approx P \times Q^T \quad (1)$$

Tabel 3.3. Penerapan ALS pada Program

```
# Dimensi laten
k = 2
num_users, num_items = R.shape

# Inisialisasi P (preferensi pengguna) dan Q (fitur item)
np.random.seed(42)
P = np.random.rand(num_users, k)
Q = np.random.rand(num_items, k)

# Hyperparameter
lambda_reg = 0.1 # Regularisasi
num_iterations = 10 # Iterasi maksimum

for iteration in range(num_iterations):
    # Update P (preferensi pengguna)
    for u in range(num_users):
        # Ambil rating pengguna ke-u
        r_u = R[u, :]
        # Ambil Q yang sesuai untuk item dengan rating
        yang_ada = Q[r_u > 0] # Hanya kolom dengan rating
        yang_diketahui = r_u[r_u > 0] # Rating yang diketahui
        if len(r_u) > 0:
            # Matriks sistem linier: (Q^T Q + lambda I) * p_u =
            Q^T r_u
            A = yang_ada.T @ yang_ada + lambda_reg * np.eye(k)
            b = yang_ada.T @ r_u
            # Dekomposisi QR
            Q_u, R_u = qr(A) # Faktorkan A
            P[u, :] = solve(R_u, Q_u.T @ b) # Pecahkan
            sistem linier

    # Update Q (fitur item)
    for i in range(num_items):
        # Ambil rating item ke-i
        r_i = R[:, i]
        # Ambil P yang sesuai untuk pengguna dengan
        rating_yang_ada = P[r_i > 0] # Hanya baris dengan rating
        yang_diketahui = r_i[r_i > 0] # Rating yang diketahui
        if len(r_i) > 0:
            # Matriks sistem linier: (P^T P + lambda I) * q_i =
            P^T r_i
            A = rating_yang_ada.T @ rating_yang_ada + lambda_reg * np.eye(k)
            b = rating_yang_ada.T @ r_i
            # Dekomposisi QR
            Q_i, R_i = qr(A) # Faktorkan A
            Q[i, :] = solve(R_i, Q_i.T @ b) # Pecahkan
            sistem linier
```

Proses pembentukan kedua matriks P dan Q diawali dengan inisialisasi nilai elemen secara acak. Langkah ini memberikan nilai awal pada elemen-elemen matriks sebelum iterasi dimulai. Selanjutnya, algoritma Alternating Least Squares (ALS) berjalan dengan cara memperbarui elemen-elemen matriks secara bergantian hingga jumlah iterasi yang telah ditentukan tercapai. Dalam implementasi ini, iterasi dilakukan sebanyak 10 kali. Pemilihan jumlah iterasi bersifat fleksibel dan dapat disesuaikan berdasarkan kebutuhan, dengan tujuan utama mencapai konvergensi. Konvergensi terjadi ketika perubahan

nilai elemen pada P dan Q menjadi sangat kecil, sehingga nilai matriks prediksi mendekati matriks sebenarnya.

```

Algorithm 1 ALS for Matrix Completion
Initialize X, Y
repeat
  for u = 1 ... n do
    
$$x_u = \left( \sum_{r_{ui} \in R_{u*}} y_i y_i^T + \lambda I_k \right)^{-1} \sum_{r_{ui} \in R_{u*}} r_{ui} y_i$$

  end for
  for i = 1 ... m do
    
$$y_i = \left( \sum_{r_{ui} \in R_{*i}} x_u x_u^T + \lambda I_k \right)^{-1} \sum_{r_{ui} \in R_{*i}} r_{ui} x_u$$

  end for
until convergence

```

Gambar 3.5. Algoritma ALS

Sumber:

<https://web.stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf>

- Perbarui nilai elemen P (preferensi pengguna)  
Untuk setiap pengguna u, baris p\_u diperbarui dengan menyelesaikan persamaan linier:

$$(Q^T Q + \lambda I) p_u = Q^T r_u, \tag{2}$$

dengan:

- Q adalah matriks fitur item
- r\_u adalah vector rating yang diberikan oleh pengguna u (hanya yang diketahui)
- λ adalah parameter regularisasi untuk mencegah overfitting.
- I adalah matriks identitas ukuran k × k.

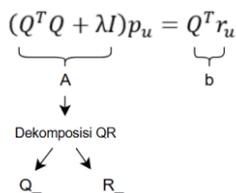
- Perbarui nilai elemen Q (fitur item)  
Untuk setiap pengguna u, baris p\_u diperbarui dengan menyelesaikan persamaan linier:

$$(P^T P + \lambda I) q_i = P^T r_i, \tag{3}$$

dengan:

- P adalah matriks preferensi pengguna
- r\_i adalah vector rating yang diberikan untuk item i (hanya yang diketahui)

Dalam penyelesaian persamaan linier, digunakan dekomposisi QR.



Gambar 3.6. Dekomposisi QR pada ALS

Sumber: Kolesi Pribadi

Berikut perhitungan yang digunakan setelah matriks difaktorkan dengan dekomposisi QR:

$$Q_R p_u = b \tag{4}$$

$$Q_-^T Q_- R_- p_u = Q_-^T b \tag{5}$$

$$R_- p_u = Q_-^T b \tag{6}$$

### C. Loss Function

Fungsi loss berperan penting dalam mengukur seberapa baik prediksi rating yang belum diketahui dalam system rekomendasi. Dalam ALS, fungsi loss dihitung berdasarkan perbedaan antara rating yang diprediksi oleh model dengan rating yang sebenarnya ada dalam data. Dalam algoritma ini, fungsi yang digunakan adalah *mean squared error* (MSE) yang diperburuk dengan penambahan regularisasi untuk mencegah overfitting.

Tabel 3.4. Loss Function pada Program

```

# Loss Function
loss = np.sum((R[R > 0] - (P @ Q.T)[R > 0]) ** 2) +
lambda_reg * (np.sum(P**2) + np.sum(Q**2))
print(f"Iteration {iteration + 1}, Loss: {loss:.4f}")

```

```

Iteration 1, Loss: 50658.3875
Iteration 2, Loss: 41979.0468
Iteration 3, Loss: 40576.1729
Iteration 4, Loss: 39885.9499
Iteration 5, Loss: 39405.2190
Iteration 6, Loss: 39046.2555
Iteration 7, Loss: 38760.4895
Iteration 8, Loss: 38537.4068
Iteration 9, Loss: 38351.4206
Iteration 10, Loss: 38182.8504

```

Gambar 3.6. Nilai Loss Function

Sumber: Kolesi Pribadi

### D. Hasil Prediksi

Matriks hasil prediksi dihitung dengan menggunakan perkalian matriks antara P (preferensi pengguna) dan transpose dari Q (fitur item). Kedua matriks tersebut telah diperbarui berdasarkan dekomposisi matriks yang telah dilakukan.

$$R \approx P \times Q^T \tag{7}$$

Hasil dari matriks prediksi menunjukkan estimasi model terhadap rating untuk setiap kombinasi pengguna dan item. Matriks ini dapat digunakan untuk membuat rekomendasi dengan cara memilih item dengan nilai prediksi tertinggi untuk setiap pengguna.

Tabel 3.7. Hasil Matriks Prediksi pada Program

```

# prediction
predicted_ratings = P @ Q.T

print("\nPredicted Ratings:")
print(predicted_ratings)

```

```

Predicted Ratings:
[[ 7.69581693  7.90231492  6.64046256 ...  5.6792221  4.55701458
  7.50488433]
 [10.15332463  8.69007867  9.41280199 ...  9.76134443  8.97650466
  9.06534546]
 [ 8.22292591  7.43910093  7.47251576 ...  7.38106626  6.58462046
  7.53506621]
 ...
 [ 9.22293429  8.47833121  8.33073671 ...  8.10284176  7.15560827
  8.51623231]
 [ 7.86079711  6.90531477  7.2208639 ...  7.32548838  6.64675974
  7.10391537]
 [ 8.38343424  6.03805834  8.19907261 ...  9.5461223  9.35392111
  6.93732179]]

```

Gambar 3.8. Matriks Prediksi

Sumber: Kolesi Pribadi

Prediksi ini memberikan gambaran sejauh mana model mampu merekonstruksi matriks rating asli berdasarkan rating

asli (data yang diketahui) menjadi dasar evaluasi kinerja model, seperti melalui perhitungan fungsi *loss*. Hasil prediksi ini penting untuk memvalidasi kualitas rekomendasi yang diberikan oleh sistem.

Tabel 3.6. Menampilkan 3 Rekomendasi Teratas pada Program

```
# Menampilkan rekomendasi berdasarkan nilai rating tertinggi
print("\nTop 3 Recommendations for Each User:")
for u in range(num_users):
    # Ambil prediksi rating untuk item yang belum diberi rating
    unrated_items = [(i, predicted_ratings[u, i]) for i in range(num_items) if R[u, i] == 0]
    # Sortir berdasarkan nilai prediksi (rating tertinggi terlebih dahulu)
    sorted_recommendations = sorted(unrated_items, key=lambda x: x[1], reverse=True)
    # Ambil hanya 3 rekomendasi teratas
    top_3_recommendations = sorted_recommendations[:3]
    print(f"\nUser {u+1}'s Top 3 Recommended Items:")
    for rank, (item_idx, rating) in enumerate(top_3_recommendations, start=1):
        print(f"Rank {rank}: Item {item_idx+1} with predicted rating {rating:.4f}")
```

```
Top 3 Recommendations for Each User:

User 1's Top 3 Recommended Items:
Rank 1: Item 45 with predicted rating 20.9266
Rank 2: Item 249 with predicted rating 13.0560
Rank 3: Item 895 with predicted rating 12.7837

User 2's Top 3 Recommended Items:
Rank 1: Item 436 with predicted rating 12.0326
Rank 2: Item 712 with predicted rating 11.3935
Rank 3: Item 8 with predicted rating 11.3588

User 3's Top 3 Recommended Items:
Rank 1: Item 45 with predicted rating 10.8334
Rank 2: Item 127 with predicted rating 9.5456
Rank 3: Item 243 with predicted rating 9.3971
```

Gambar 3.9. Output Program  
Sumber: Koleksi Pribadi

## VI. KESIMPULAN

Penerapan dekomposisi QR dalam sistem rekomendasi berbasis *Collaborative Filtering* dapat meningkatkan akurasi prediksi, terutama ketika dihadapkan pada data yang bersifat jarang (*sparse*). Dengan mengintegrasikan metode *Alternating Least Square* (ALS) sebagai kerangka utama untuk memfaktorkan matriks interaksi pengguna-item dan menggunakan dekomposisi QR sebagai alternatif dari dekomposisi SVD, proses penyelesaian sistem persamaan menjadi lebih stabil dan efisien.

Dekomposisi QR memungkinkan pembentukan matriks orthogonal dan matriks segitiga atas yang tidak hanya menjaga kestabilan perhitungan tetapi juga meningkatkan akurasi prediksi dengan mengurangi efek dari data yang tidak lengkap (*sparse*). Sementara itu, ALS berfungsi untuk memperkirakan nilai-nilai matriks yang hilang dengan cara meminimalkan kesalahan kuadrat rata-rata (*mean squared error*/MSE) antara matriks asli dan hasil perkiraan dari perkalian matriks pengguna ( $U$ ) dan matriks fitur item ( $V^T$ ).

Pendekatan ini tidak hanya memperbaiki kualitas prediksi tetapi juga memberikan solusi yang efisien untuk menangani

dataset besar. Dengan dekomposisi QR dapat memberikan kontribusi signifikan dalam memastikan bahwa model rekomendasi tetap andal meskipun data yang digunakan memiliki banyak elemen kosong. Kombinasi kedua metode ini menunjukkan potensi besar untuk diterapkan dalam berbagai aplikasi sistem rekomendasi, seperti rekomendasi film, buku, atau produk *e-commerce*.

## VII. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas rahmat dan bimbingan-Nya, penulis dapat menyelesaikan makalah ini yang berjudul "*Penerapan Dekomposisi QR dalam Meningkatkan Akurasi Sistem Rekomendasi Berbasis Collaborative Filtering*" tepat waktu. Makalah ini disusun sebagai bagian dari tugas pada mata kuliah Aljabar Linier dan Geometri IF2123.

Penulis ingin menyampaikan rasa terima kasih yang mendalam kepada Bapak Dr. Rila Mandala, Dr. Ir. Rinaldi, M.T., Bapak Dr. Judhi Santoso, M.Sc., dan Bapak Arrival Dwi Sentosa, M.T., yang dengan tulus dan penuh dedikasi telah membimbing dan membagikan ilmu selama perkuliahan berlangsung. Ilmu yang diberikan menjadi bekal berharga dalam proses penyelesaian makalah ini.

Tidak lupa, penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua yang selalu memberikan dukungan, baik secara moral maupun materi, selama proses pengerjaan makalah ini. Ucapan terima kasih juga penulis sampaikan kepada teman-teman dekat—Adinda Putri, Ranashahira, dan Wardatul Khoiroh—yang telah menjadi tempat berbagi ide, memberikan semangat, dan selalu mendukung selama proses ini berlangsung.

Penulis menyadari bahwa makalah ini masih jauh dari sempurna. Oleh karena itu, penulis memohon maaf apabila terdapat kesalahan atau kekurangan dalam penulisan ini. Semoga makalah ini dapat memberikan manfaat bagi para pembaca dan menjadi langkah kecil yang berkontribusi pada pengembangan pengetahuan.

## REFERENCES

- [1] MTI Binus, "Sistem Rekomendasi Content-Based," [Online]. Tersedia: <https://mti.binus.ac.id/2020/11/17/sistem-rekomendasi-content-based/>. [Diakses pada 27 Desember 2024].
- [2] NVIDIA, "Recommendation System Glossary," [Online]. Tersedia: <https://www.nvidia.com/en-us/glossary/recommendation-system/>. [Diakses pada 27 Desember 2024].
- [3] Munir, Rinaldi, "Dekomposisi QR," [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/Algeo-23b-Dekomposisi-QR-2024.pdf>. [Diakses pada 27 Desember 2024].
- [4] Aku Pintar, "Matriks: Pengertian, Operasi, Determinan, Invers, dan Contoh Soal," [Online]. Tersedia: <https://akupintar.id/info-pintar/-/blogs/matriks-pengertian-operasi-determinan-invers-dan-contoh-soal>. [Diakses pada 28 Desember 2024].
- [5] *GeeksforGeeks*, "Matrix Operations," [Online]. Tersedia: <https://www.geeksforgeeks.org/matrix-operations/>. [Diakses pada 28 Desember 2024].
- [6] *GeeksforGeeks*, "What are Recommender Systems?," [Online]. Tersedia: <https://www.geeksforgeeks.org/what-are-recommender-systems/>. [Diakses pada 28 Desember 2024].
- [7] *Towards Data Science*, "Prototyping a Recommender System Step-by-Step Part 2: Alternating Least Square (ALS) Matrix," [Online]. Tersedia: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-als-matrix/>.

by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1.  
[Diakses pada 29 Desember 2024].

- [8] Stanford University, "CME323: Iterative Methods for Least Squares," [Online]. Tersedia: <https://web.stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf>. [Diakses pada 29 Desember 2024].

### 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, 30 Desember 2024



Heleni Gratia M Tampubolon  
(13523107)