



PENGEMBANGAN SISTEM *BACKEND REKOMENDASI PRODUK DENGAN COLLABORATIVE FILTERING BERBASIS SINGULAR VALUE DECOMPOSITION*

Ahmad Waluyo¹, Tifanny Nabarian², Lukman Rosyidi³

^{1,2,3}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

Jakarta Selatan, DKI Jakarta, Indonesia 12640

ahma22291ti@student.nurulfikri.ac.id, nabarian@nurulfikri.ac.id, lukman@nurulfikri.ac.id

Abstract

Recommendation systems have become a crucial component in enhancing user experience on e-commerce platforms, including PT Renos Marketplace Indonesia. However, the existing recommendation system was still random and unable to adapt to user preferences. This study aims to develop a product recommendation backend system based on Collaborative Filtering using the Singular Value Decomposition (SVD) algorithm. The methodology employed is Research and Development (R&D) with an Agile Scrum approach, along with performance evaluation using RMSE, MSE, and MAE metrics. The system was developed using FastAPI, PostgreSQL, and Python, and its functionality was tested using black-box testing methods. The implementation results show an RMSE of 0.1886, MSE of 0.0366, and MAE of 0.1242, indicating excellent prediction accuracy. Additionally, a user perception survey involving 11 internal respondents showed an average satisfaction score above 80%, with the highest score of 90.91% for the statement indicating that the system increases purchase likelihood. These findings demonstrate that the SVD algorithm is effective in generating relevant and personalized recommendations. The study concludes that the developed backend recommendation system successfully improves the efficiency and relevance of product recommendations and opens opportunities for further development through the integration of more complex user behavior data.

Keywords: Backend System, Collaborative Filtering, E-commerce, Recommendation System, Singular Value Decomposition

Abstrak

Sistem rekomendasi telah menjadi komponen penting dalam meningkatkan pengalaman pengguna di *platform e-commerce*, termasuk di PT. Renos Marketplace Indonesia. Namun, sistem rekomendasi sebelumnya masih bersifat acak dan belum mampu menyesuaikan dengan preferensi pengguna. Penelitian ini bertujuan mengembangkan sistem *backend* rekomendasi produk berbasis *Collaborative Filtering* menggunakan algoritma *Singular Value Decomposition* (SVD). Metodologi yang digunakan adalah *Research and Development* (R&D) dengan pendekatan *Agile Scrum* serta evaluasi performa menggunakan metrik RMSE, MSE, dan MAE. Sistem dibangun menggunakan FastAPI, PostgreSQL, dan Python, serta diuji secara fungsional melalui metode *blackbox testing*. Hasil implementasi menunjukkan nilai RMSE sebesar 0,1886; MSE sebesar 0,0366; dan MAE sebesar 0,1242 yang menandakan akurasi prediksi sangat baik. Selain itu, hasil survei terhadap 11 responden internal menunjukkan rata-rata kepuasan di atas 80%, dengan skor tertinggi sebesar 90,91% pada pernyataan bahwa sistem meningkatkan peluang pembelian. Temuan ini menunjukkan bahwa algoritma SVD efektif dalam menghasilkan rekomendasi yang relevan dan personal. Penelitian ini menyimpulkan bahwa sistem *backend* rekomendasi yang dikembangkan berhasil meningkatkan efisiensi dan relevansi rekomendasi, serta membuka peluang pengembangan lebih lanjut dengan integrasi perilaku pengguna yang lebih kompleks.

Kata kunci: Collaborative Filtering, E-commerce, Singular Value Decomposition, Sistem Backend, Sistem Rekomendasi

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi besar dalam cara manusia melakukan aktivitas ekonomi, khususnya dalam sektor perdagangan [1]. Proses jual beli yang dulunya mengandalkan interaksi langsung

kini banyak beralih ke sistem digital melalui *platform e-commerce* [2]. *Platform* ini menjadi sarana yang sangat membantu, terutama bagi pelaku Usaha Mikro, Kecil, dan Menengah (UMKM), dalam memasarkan produk sekaligus memantau kinerja penjualannya secara efisien [3]. Dengan

adanya *e-commerce*, hambatan seperti keterbatasan tempat usaha atau proses perizinan pun dapat diminimalkan.

Pertumbuhan *e-commerce* di Asia Tenggara menunjukkan tren yang sangat positif, didorong oleh meningkatnya akses terhadap internet dan pertumbuhan kelas menengah yang signifikan [4]. Di Indonesia sendiri, pasar *e-commerce* diprediksi akan terus mengalami ekspansi. Menurut data Statista, jumlah pengguna *e-commerce* di Indonesia diperkirakan akan tumbuh lebih dari dua kali lipat, dari 33,5 juta pada tahun 2024 menjadi 99,1 juta pada 2029 [5]. Pertumbuhan ini memicu munculnya berbagai *startup* lokal yang turut meramaikan ekosistem digital tersebut.

Salah satu *startup* yang tengah berkembang adalah PT. Renos Marketplace Indonesia, atau dikenal sebagai Renos.id. Didirikan pada tahun 2021, Renos.id merupakan *platform e-commerce* yang bergerak di sektor *home-living* dengan model bisnis *business-to-consumer* (B2C). Perusahaan ini merupakan bagian dari ekspansi *platform* nocnoc.com yang berbasis di Thailand, yang juga merupakan anak perusahaan dari Siam Cement Group [6].

Seiring bertambahnya jumlah pengguna dan produk yang tersedia, Renos.id menghadapi tantangan baru dalam hal penyajian konten yang relevan bagi pengguna. Salah satu isu utama adalah bagaimana memilih dan menampilkan produk yang tepat di halaman rekomendasi agar pengalaman pengguna menjadi lebih personal. Saat ini, sistem rekomendasi yang digunakan masih bersifat statis dan acak (*hardcoded*), sehingga efektivitasnya dalam meningkatkan penjualan dinilai kurang optimal.

Untuk mengatasi tantangan tersebut, diperlukan pendekatan teknologi yang lebih canggih, seperti implementasi sistem rekomendasi berbasis *machine learning*. Penelitian menunjukkan bahwa sistem rekomendasi yang efektif mampu meningkatkan kepuasan pelanggan secara signifikan, dengan indeks kepuasan (*Customer Satisfaction Index/CSI*) mencapai angka 97,12% pada *platform e-commerce* yang menerapkannya [7]. Hal ini menunjukkan adanya hubungan positif yang kuat antara sistem rekomendasi dan loyalitas pengguna.

Dalam konteks ini, PT. Renos Marketplace Indonesia mengembangkan sistem *backend* rekomendasi produk menggunakan metode *Collaborative Filtering* yang diperkuat dengan algoritma *Singular Value Decomposition* (SVD). SVD dipilih karena terbukti menghasilkan rekomendasi yang lebih akurat jika dibandingkan dengan algoritma lain seperti *K-Nearest Neighbors* [8]. Pengujian dan evaluasi sistem juga menunjukkan bahwa SVD memiliki performa unggul dibanding metode lain seperti *K-Means*, menjadikannya sebagai kandidat utama dalam pengembangan sistem ini.

Proses pengembangan sistem dilakukan menggunakan pendekatan *Agile Scrum*, yang memungkinkan pembaruan dan penyesuaian secara berkala berdasarkan umpan balik

dari pengguna. Metode ini memberikan fleksibilitas dalam proses iterasi dan memastikan bahwa sistem yang dibangun dapat menyesuaikan dengan kebutuhan dinamis pasar. Melalui pengembangan sistem rekomendasi ini, Renos.id berharap dapat meningkatkan efektivitas penjualan, memperkuat loyalitas pelanggan, serta meningkatkan *Gross Merchandise Value* (GMV) secara keseluruhan.

Penelitian mengenai sistem rekomendasi berbasis SVD telah banyak dilakukan untuk meningkatkan akurasi dan efisiensi dalam menyajikan rekomendasi yang relevan. Penelitian yang dilakukan oleh Sitorus mengimplementasikan algoritma SVD pada sistem rekomendasi film menggunakan *dataset* dari Kaggle, dan menghasilkan performa yang baik dengan nilai RMSE sebesar 0,4002 serta MAE sebesar 0,1186 menunjukkan bahwa SVD mampu memberikan prediksi *rating* yang lebih rendah tingkat kesalahannya [9]. Di sisi lain, Ding et al. [10] mengembangkan pendekatan *Collaborative Filtering* berbasis PCA yang berhasil meningkatkan efisiensi dan akurasi sistem rekomendasi pada *dataset* MovieLens dengan skala besar, yaitu 20 juta data *rating*.

Sementara itu, Kabić dkk. [11] menyempurnakan metode SVD dengan menambahkan proses inisialisasi berbasis *K-means clustering*, yang terbukti mampu meningkatkan kinerja model dan mengungguli pendekatan-pendekatan sebelumnya. Di penelitian lain, Hong dkk. [12] mengusulkan model SVD-AE, yakni integrasi SVD dengan *autoencoder* sederhana, yang terbukti mampu meningkatkan efisiensi sistem dan ketahanan terhadap *noise* pada *dataset* Gowalla dan Yelp18. Rangkaian studi tersebut menunjukkan bahwa pendekatan *model-based collaborative filtering*, khususnya yang melibatkan modifikasi algoritma SVD, memiliki potensi signifikan dalam pengembangan sistem rekomendasi yang lebih cerdas dan responsif.

Meskipun berbagai studi telah menunjukkan keunggulan algoritma SVD dalam sistem rekomendasi, masih sedikit penelitian yang secara khusus mengembangkan dan mengukur efektivitas sistem *backend* rekomendasi berbasis SVD dalam konteks *e-commerce* lokal seperti renos.id. Kesenjangan inilah yang menjadi dasar penting dilakukannya penelitian ini. Penelitian ini bertujuan untuk membangun sistem *backend* rekomendasi produk yang lebih personal dengan metode *Collaborative Filtering* berbasis SVD, serta mengukur efektivitasnya dalam meningkatkan relevansi produk yang ditampilkan. Melalui penelitian ini, diharapkan dapat memberikan kontribusi keilmuan dalam implementasi algoritma SVD pada sistem rekomendasi, mendukung renos.id dalam meningkatkan personalisasi layanan dan konversi penjualan, serta memberikan pengalaman belanja yang lebih relevan dan efisien bagi pengguna.

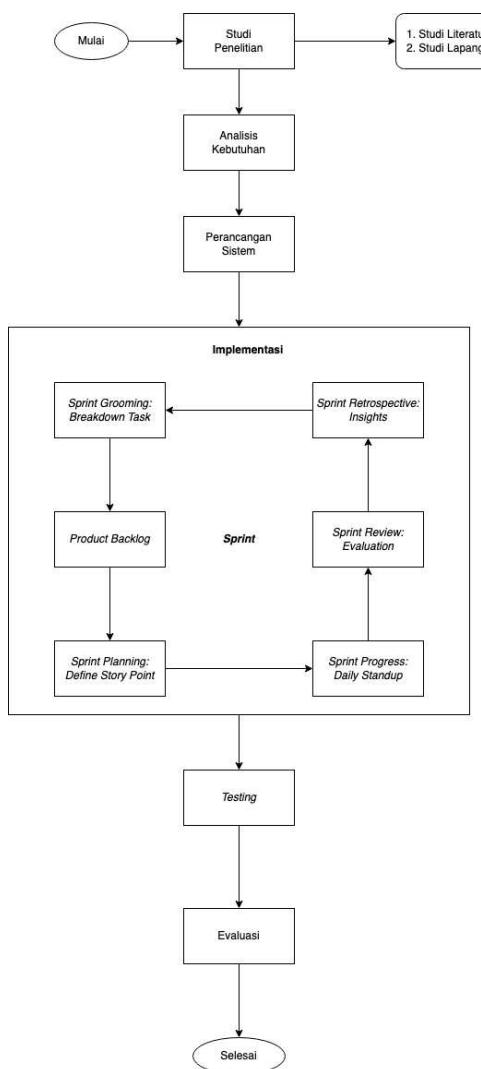
2. METODE PENELITIAN

2.1 Jenis Penelitian

Penelitian ini menggunakan metode *Research and Development* (R&D) yang bertujuan untuk mengembangkan dan mengimplementasikan sistem *backend* rekomendasi produk berbasis *Collaborative Filtering* dengan algoritma SVD di PT. Renos Marketplace Indonesia. Pendekatan R&D mencakup proses identifikasi masalah pada sistem rekomendasi sebelumnya yang masih bersifat acak, pengembangan model *machine learning*, serta evaluasi performa sistem secara iteratif. Melalui metode ini, diharapkan dapat dihasilkan sistem yang lebih akurat dan relevan dengan preferensi pengguna, sehingga mampu meningkatkan kepuasan pelanggan serta efektivitas bisnis perusahaan.

2.2 Tahapan Penelitian

Penelitian ini terdiri dari enam tahapan utama seperti pada Gambar 1, yaitu studi penelitian, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi.



Gambar 1. Tahapan Penelitian

Tahapan dimulai dengan studi literatur dan studi lapangan untuk memahami teori serta praktik sistem rekomendasi berbasis *Collaborative Filtering* menggunakan algoritma SVD. Studi literatur dilakukan terhadap konsep sistem rekomendasi, struktur *backend*, dan metode pengembangan *Agile Scrum*, sementara studi lapangan dilakukan dengan mengamati fitur rekomendasi di *platform e-commerce* seperti Tokopedia. Tahap selanjutnya adalah analisis kebutuhan, yang bertujuan mengidentifikasi kebutuhan fungsional dan non-fungsional sistem, termasuk arsitektur *backend*, standar keamanan, dan spesifikasi fitur utama. Hasil analisis ini digunakan dalam tahap perancangan sistem, mencakup pembuatan struktur *database*, *Application Programming Interface* (API), *Entity Relationship Diagram* (ERD), serta alur data dari proses *input* pengguna hingga penyajian rekomendasi produk. Tahap implementasi dilakukan menggunakan metode *Agile Scrum* yang dibagi ke dalam *Sprint*, meliputi *Product Backlog*, *Sprint Planning*, *Sprint Review*, hingga *Sprint Retrospective* untuk memastikan proses berjalan efisien dan responsif terhadap umpan balik. Setelah sistem diimplementasikan, dilakukan pengujian menggunakan metode *black-box testing* untuk memastikan fitur *backend* bekerja sesuai dengan fungsinya, termasuk validasi akurasi rekomendasi. Tahap akhir adalah evaluasi terhadap metodologi, keberhasilan implementasi, dan efektivitas sistem yang dikembangkan, serta penyusunan rekomendasi untuk pengembangan selanjutnya. Evaluasi ini mencakup analisis kesesuaian metode, kendala teknis, performa sistem, dan potensi peningkatan algoritma maupun infrastruktur sistem di masa mendatang.

2.3 Lokasi dan Perangkat Penelitian

Penelitian dilaksanakan di PT. Renos Marketplace Indonesia yang berlokasi di Menara Rajawali, Jl. DR. Ide Anak Agung Gde Agung No.Lot #5.1, Kuningan, Jakarta Selatan. Seluruh proses pengembangan dan pengujian sistem dilakukan menggunakan layanan *Virtual Private Server* (VPS), yang terdiri dari dua lingkungan terpisah *production* dan *staging* untuk mengakomodasi kebutuhan manajemen sumber daya dan simulasi *end-user* secara optimal. Lingkungan produksi memiliki kapasitas CPU dan RAM lebih besar karena menangani data pengguna yang lebih kompleks, sedangkan *staging* digunakan untuk pengujian awal. Akses ke VPS dilakukan menggunakan laptop MacBook Air M2 (2022) dengan spesifikasi CPU Apple Silicon M2, RAM 8 GB, dan penyimpanan 512 GB SSD. Dalam mendukung pengembangan, penulis juga menggunakan berbagai perangkat lunak seperti Visual Studio Code, Docker Desktop, DBeaver, Jupyter Notebook, Star UML, yang berfungsi untuk *scripting*, manajemen *database*, pemodelan sistem, hingga penyusunan laporan penelitian secara sistematis.

2.4 Metode Pengumpulan dan Analisis Data

Penelitian ini menggunakan pendekatan kuantitatif untuk menganalisis kinerja sistem *backend* rekomendasi produk berbasis *Collaborative Filtering* dengan algoritma SVD, dengan pengumpulan data melalui studi dokumen, observasi langsung, dan data internal dari PT. Renos Marketplace Indonesia. Studi dokumen mencakup analisis kebutuhan, spesifikasi desain, dan catatan pengujian; sedangkan observasi dilakukan terhadap proses pengembangan dan pengujian sistem di lingkungan *staging* serta praktik rekomendasi di *e-commerce* lain sebagai

pembanding. Data utama berasal dari *database* internal perusahaan berupa data *dummy* pada lingkungan *staging* yang merepresentasikan riwayat transaksi dan interaksi pengguna. Data ini kemudian dianalisis menggunakan metrik evaluasi seperti *Root Mean Squared Error* (RMSE), *Mean Squared Error* (MSE), dan *Mean Absolute Error* (MAE), serta tingkat keberhasilan interaksi pengguna terhadap produk yang direkomendasikan. Melalui kombinasi metode ini, penelitian diharapkan mampu memberikan gambaran menyeluruh mengenai efektivitas sistem rekomendasi yang dikembangkan.

Tabel 1. Test Case

No	Instrumen Pengujian	Input	Output yang Diharapkan	Hasil Uji	Status
1	Pengujian rekomendasi produk	Query Param: { "total": 30 }	Kode 200, status <i>success</i> , data mencakup <i>total recommendations</i> , <i>avg_rmse</i> , <i>avg_mse</i> , <i>avg_mae</i> , dan <i>dateup</i>	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
2	Validasi rekomendasi produk di <i>database</i>	Query Param: { "total": 30 }	Data rekomendasi yang tersimpan sesuai dengan data yang dikembalikan oleh <i>endpoint</i>	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
3	Pengujian jumlah rekomendasi produk berdasarkan parameter "total"	Query Param: { "total": 30 }	Data 30 rekomendasi produk untuk setiap pengguna tersimpan di DB	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
4	Pengujian respons API terhadap <i>input</i> total tidak valid (<i>string</i>)	Query Param: { "total": "test" }	Kode 422, status <i>validation_error</i> , message: "Input should be a valid integer, unable to parse string as an integer"	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
5	Pengujian respons API terhadap <i>input</i> total tidak valid (kurang dari 6 produk)	Query Param: { "total": 4 }	Kode 422, status <i>validation_error</i> , message: "params total must be at least 6"	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
6	Pengujian penyimpanan data pemrosesan produk	Tidak ada <i>input</i>	Kode 200, status <i>success</i> , message: "Successfully created table processed_product"	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
7	Pengujian kontrol parameter bobot (normal)	Body: { "weight_purchased": 0.7, "weight_rating": 0.9, "weight_favorite": 0.5, "weight_view": 0.1 }	Kode 200, status <i>success</i> , message: "product recommendation parameter has been updated successfully"	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
8	Pengujian kontrol parameter bobot (salah: nilai <= 0 atau nilai >1)	Body: { "weight_purchased": 0, "weight_rating": 1.9, "weight_favorite": 0.5, "weight_view": 0.1 }	Kode 422, status <i>validation_error</i> , message yang menjelaskan bahwa nilai <i>weight</i> tidak boleh kurang dari 0 atau tidak lebih dari 1	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]
9	Integrasi <i>database</i> dari replika ke <i>main database</i>	Tidak ada <i>input</i>	Kode 201, status <i>success</i> , message: "Data Integrated Successfully"	[Hasil sesuai/tidak sesuai]	[Lulus / Gagal]

2.5 Metode Pengujian dan Evaluasi

Pengujian dan evaluasi merupakan tahap penting untuk memastikan bahwa sistem *backend* rekomendasi produk yang dikembangkan dapat berfungsi sesuai harapan serta memberikan hasil yang relevan bagi pengguna [13]. Pengujian dilakukan untuk mengecek kinerja teknis dan

fungsionalitas sistem berbasis API, sementara evaluasi bertujuan menilai efektivitas algoritma rekomendasi dalam konteks pengalaman pengguna. Tahapan ini menggunakan pendekatan terstruktur untuk menilai baik *output* sistem secara logis maupun persepsi pengguna terhadap kualitas rekomendasi yang diberikan.

Pada tahap pengujian, metode yang digunakan adalah *black-box testing*, yang berfokus pada fungsionalitas sistem tanpa melihat kode internal [14]. Pengujian dilakukan dengan berbagai skenario *input* dan validasi *output* melalui *endpoint API*, seperti pengiriman parameter total, kontrol parameter bobot, hingga validasi respons terhadap *input* yang tidak valid. Hasil respons sistem kemudian dikonfirmasi melalui pengecekan ke *database* untuk memastikan konsistensi data yang direkomendasikan. Pendekatan ini dipilih karena efektif dalam menilai apakah sistem telah berjalan sesuai kebutuhan pengguna, tanpa harus mengetahui struktur di dalamnya. Sementara itu, evaluasi dilakukan dengan dua pendekatan: pengukuran metrik akurasi sistem (RMSE, MSE, MAE) dan survei pengguna menggunakan Skala *Likert* 1-5. Skor dihitung menggunakan rumus persentase *Likert* berdasarkan total nilai aktual dibanding skor ideal, untuk menilai persepsi pengguna terhadap relevansi hasil rekomendasi [15].

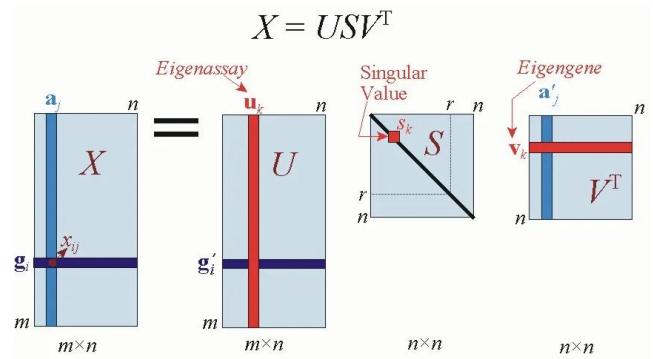
Evaluasi ini bertujuan untuk mengidentifikasi kekuatan dan kelemahan sistem, serta memberikan *insight* perbaikan. Hasil evaluasi ditampilkan pada Tabel 1 yang diklasifikasikan dalam status berhasil, gagal, atau perlu perbaikan, berdasarkan akurasi prediksi dan kepuasan pengguna. Temuan dari evaluasi juga menjadi dasar untuk pengembangan lebih lanjut agar sistem dapat memberikan rekomendasi yang lebih personal dan sesuai preferensi pengguna.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

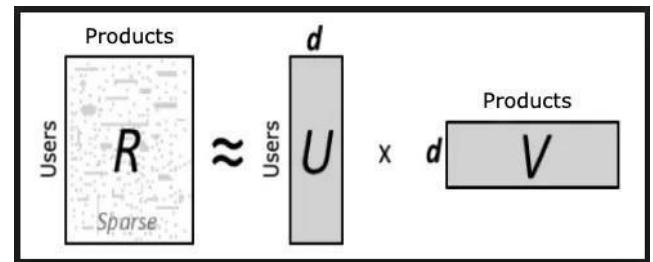
Perancangan sistem rekomendasi produk ini menggunakan *model-based collaborative filtering* dan menggunakan algoritma SVD. Pendekatan *Collaborative Filtering* dipilih setelah penulis melakukan studi lapangan dimana mayoritas sistem rekomendasi produk sering kali menampilkan item yang populer seperti mendapatkan *rating* tinggi ataupun produk yang banyak dibeli. Oleh karenanya, butuh kolaborasi antar pengguna-item untuk mewujudkan konsep filterisasi produk seperti ini. Namun demikian metode *Collaborative Filtering* masih memiliki kekurangan seperti *cold start* dan *data sparsity* [16]. Oleh karenanya butuh algoritma yang mampu menjawab permasalahan ini sehingga rekomendasi produk yang dihasilkan dapat lebih optimal. Dalam hal ini SVD sangat tepat karena kemampuannya dalam menangkap pola tersembunyi pada matriks pengguna-item dan dapat memprediksi nilai *sparse* pada *item rating* yang belum dinilai oleh pengguna [17].

Dalam konteks Aljabar Linier, SVD merupakan teknik untuk mendekomposisi sebuah matriks ke dalam tiga komponen utama, yaitu dua matriks ortogonal (*user matrix* dan *item matrix*) dan satu matriks diagonal (*singular values matrix*) [18]. Rumus selengkapnya seperti pada Gambar 2 berikut.



Gambar 2. Rumus Singular Value Decomposition

Proses SVD seperti pada Gambar 2 menjelaskan dekomposisi matriks X yang dibagi menjadi tiga matriks: U , S , dan V^T . Matriks X berukuran $m \times n$ mewakili data asli, misalnya interaksi pengguna dan produk, di mana baris merepresentasikan pengguna dan kolom merepresentasikan produk atau *item*. Matriks U ($m \times r$) berisi representasi laten dari pengguna, S adalah matriks diagonal ($r \times r$) yang memuat *singular values* sebagai bobot pentingnya dimensi laten, dan V^T ($r \times n$) berisi representasi laten dari produk. Dengan mengalikan kembali ketiga matriks ini, sistem dapat merekonstruksi data asli dan memprediksi nilai-nilai yang hilang dalam kondisi *sparsity*, menjadikan SVD sangat efektif dalam sistem rekomendasi berbasis model (*model-based collaborative filtering*) [19]. Berikutnya korelasi matriks SVD terhadap konteks rekomendasi produk digambarkan pada Gambar 3 berikut.

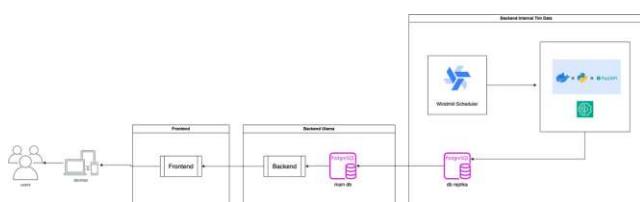


Gambar 3. User-Item Interaction Matrix

Dengan menerapkan pendekatan *model-based*, sistem mampu mempelajari pola-pola interaksi yang tersembunyi di dalam data historis dan menghasilkan representasi laten dari pengguna dan produk. Pendekatan ini sangat efektif dalam menangani permasalahan umum dalam teknik *Collaborative Filtering* yakni *data sparsity*, suatu kondisi ketika sebagian besar nilai dalam matriks interaksi tidak terisi karena kurangnya interaksi antara *user* dengan produk [20]. Dengan penerapan SVD, model dapat memprediksi nilai *rating* yang hilang berdasarkan pola dari data yang tersedia. Setelah proses pemodelan selesai kemudian hasil rekomendasi disimpan dalam basis data dan disajikan melalui *endpoint* berbasis REST API, sehingga dapat diakses secara efisien oleh sistem *frontend* atau layanan lainnya. Arsitektur sistem dibangun secara modular agar dapat menangani *input* parameter dari

pengguna, mengolahnya melalui model rekomendasi, dan menghasilkan daftar produk yang relevan. Lingkungan pengembangan dibagi menjadi dua yaitu *staging* dan *production*, guna memisahkan proses pengujian dengan implementasi sistem secara langsung. Penggunaan *Virtual Private Server* (VPS) memungkinkan pengujian dilakukan secara terpisah tanpa mengganggu infrastruktur utama.

Sistem ini beroperasi secara *batch* dengan jadwal eksekusi berkala melalui mekanisme *scheduler*, bukan dalam bentuk *real-time recommendation engine*. Proses rekomendasi mencakup beberapa tahap, mulai dari *preprocessing* data pengguna dan produk, pembentukan matriks interaksi, dekomposisi matriks menggunakan SVD, hingga menghasilkan rekomendasi dengan mempertimbangkan bobot interaksi *user-item* seperti data pembelian produk, *rating* produk, produk favorit, dan berapa banyak produk ditampilkan. Hasil dari proses ini disimpan dalam *database* replikasi untuk memisahkan beban antara sistem *Online Transaction Processing* (OLTP) pada *database* utama dan kebutuhan *Online Analytical Processing* (OLAP) untuk analisis dan pemrosesan rekomendasi, sehingga ketika terjadi permasalahan pada *database* replika operasional *database* utama tidak terkena dampaknya. Selanjutnya hasil rekomendasi disinkronisasi ke *database* utama untuk ditampilkan melalui *endpoint API* yang dapat diakses oleh *frontend* atau layanan lain. Arsitektur lengkap sistem ditampilkan pada Gambar 4 yang menunjukkan hubungan antara komponen sistem, *database*, serta *interface API* yang digunakan dalam pengujian dan pengambilan data rekomendasi.



Gambar 4. Arsitektur Sistem Rekomendasi Produk

Metode *blackbox testing* digunakan dalam penelitian ini untuk mengevaluasi fungsionalitas sistem *backend* rekomendasi produk yang dikembangkan. Pengujian dilakukan tanpa menelusuri kode sumber, melainkan dengan mengamati *input* dan *output* dari sistem melalui serangkaian permintaan API yang telah dirancang sebelumnya. Fokus utama pengujian adalah pada validasi *output* berdasarkan parameter *input* seperti jumlah rekomendasi, format data yang dihasilkan, serta tanggapan sistem terhadap *input* yang tidak valid. Setiap skenario pengujian diuji untuk memastikan sistem merespons secara benar dan sesuai spesifikasi.

Tabel 2. Rangkuman Hasil *Black-box Testing*

No	Nama Uji	Output yang Diharapkan	Hasil
1	Pengujian rekomendasi produk	Kode 200, status <i>success</i> , mencakup <i>total recommendations</i> , <i>avg_rmse</i> , <i>avg_mse</i> , <i>avg_mae</i> , dan <i>dateup</i>	Sesuai
2	Validasi rekomendasi produk di <i>database</i>	Data rekomendasi yang tersimpan sesuai dengan hasil API	Sesuai
3	Pengujian jumlah rekomendasi berdasarkan parameter “total”	30 data tersimpan di <i>database</i> per pengguna	Sesuai
4	Pengujian <i>input total</i> tidak valid (<i>string</i>)	Kode 422, status <i>validation_error</i> , pesan <i>error</i>	Sesuai
5	Pengujian <i>input total</i> kurang dari 6	Kode 422, status <i>validation_error</i> , pesan <i>error</i>	Sesuai
6	Pengujian penyimpanan data pemrosesan produk	Kode 200, status <i>success</i> , message sukses membuat tabel <i>processed_product</i>	Sesuai
7	Pengujian kontrol parameter bobot (normal)	Kode 200, status <i>success</i> , pesan sukses update parameter bobot	Sesuai
8	Pengujian kontrol parameter bobot (salah: nilai 0 atau >1)	Kode 422, status <i>validation_error</i> , pesan <i>error</i>	Sesuai
9	Integrasi <i>database</i> dari replika ke <i>main database</i>	Kode 201, status <i>success</i> , pesan sukses integrasi	Sesuai
10	Validasi <i>endpoint</i> yang tidak terdaftar	Kode 404, status <i>failed</i> , pesan <i>route</i> tidak ditemukan	Sesuai

Pengujian pada Tabel 2 dilakukan terhadap sepuluh skenario yang mewakili fitur inti dari sistem rekomendasi, termasuk *generate* rekomendasi, validasi jumlah data, pengujian parameter bobot, hingga validasi terhadap *endpoint* yang tidak terdaftar. Hasil pengujian menunjukkan bahwa sebagian besar fitur berhasil dijalankan sesuai dengan ekspektasi, dengan status pengujian “Lulus”. Salah satu uji penting adalah validasi parameter “total” yang menunjukkan bahwa sistem dapat menyesuaikan jumlah rekomendasi sesuai permintaan pengguna, dan dapat menangani kesalahan *input* seperti memasukkan nilai *string* atau angka di bawah ambang batas minimum.

Hal ini dilakukan untuk mengurangi potensi *sparsity* atau kekosongan data pada *user-item interaction matrix*, yaitu matriks yang merepresentasikan interaksi antara pengguna dan produk dalam bentuk nilai seperti *rating* atau preferensi, yang menjadi elemen utama dalam penerapan metode SVD. Pengujian terhadap kontrol bobot menunjukkan bahwa validasi *server* terhadap nilai *input* berjalan dengan baik, dengan sistem menolak nilai yang tidak berada dalam rentang 0–1. Pengujian integrasi antara *database* replika dan utama, serta validasi *endpoint* yang tidak dikenal, juga berhasil dijalankan, membuktikan bahwa sistem memiliki ketahanan terhadap kesalahan dan mampu mengelola integrasi data antar layanan.

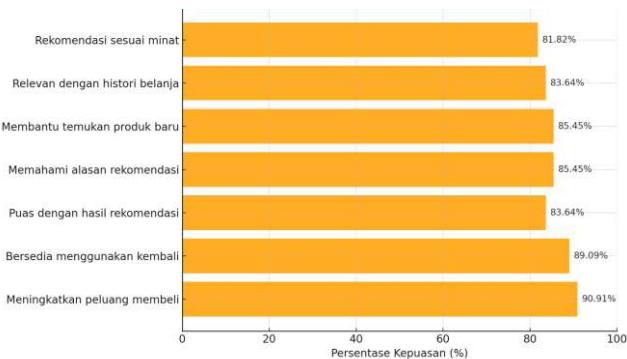
Untuk menilai efektivitas dan akurasi sistem *backend* rekomendasi produk berbasis *Collaborative Filtering* menggunakan algoritma SVD, dilakukan evaluasi menggunakan tiga metrik utama: RMSE, MSE, dan MAE. Ketiga metrik ini dipilih karena mampu menggambarkan seberapa dekat prediksi sistem terhadap nilai *rating* aktual yang seharusnya diberikan oleh pengguna terhadap suatu produk [21].

Tabel 3. Hasil Evaluasi Algoritma SVD

No	Nama Metrik	Nilai Evaluasi
1	RMSE	0.1886
2	MSE	0.0366
3	MAE	0.1242

Berdasarkan hasil evaluasi yang ditampilkan pada Tabel 3, algoritma SVD yang diterapkan menunjukkan performa prediksi yang cukup baik dengan nilai RMSE sebesar 0,1886; MSE sebesar 0,0366; dan MAE sebesar 0,1242. Nilai-nilai ini berada pada kisaran yang rendah, yang menunjukkan bahwa perbedaan antara prediksi sistem dan nilai aktual relatif kecil. Nilai RMSE yang rendah mencerminkan bahwa sistem jarang melakukan kesalahan prediksi besar, sementara MAE mengindikasikan bahwa rata-rata kesalahan prediksi juga cukup kecil. Dengan kata lain, sistem rekomendasi mampu menyarankan produk dengan tingkat relevansi yang tinggi terhadap preferensi pengguna, dan berpotensi meningkatkan pengalaman belanja pengguna di *platform* Renos.id.

Untuk mengevaluasi sejauh mana sistem rekomendasi produk yang dikembangkan relevan dengan preferensi pengguna, dilakukan survei dengan menggunakan metode Skala *Likert*. Skala yang digunakan mencakup lima tingkatan: Sangat Setuju (5), Setuju (4), Netral (3), Tidak Setuju (2), dan Sangat Tidak Setuju (1). Tujuan dari survei ini adalah memperoleh masukan kualitatif yang mendukung evaluasi kuantitatif sebelumnya.



Gambar 5. Rangkuman Hasil Responden

Gambar 5 menunjukkan ringkasan hasil tanggapan dari 11 responden karyawan PT. Renos Marketplace Indonesia terhadap sistem rekomendasi produk yang dikembangkan. Seluruh pernyataan dalam survei memperoleh skor rata-rata di atas 4 atau lebih dari 80%, yang menandakan tingkat

kepuasan pengguna berada dalam kategori "Sangat Baik". Pernyataan dengan skor tertinggi adalah nomor 7 yang mengindikasikan bahwa sistem rekomendasi dapat meningkatkan peluang pembelian produk dalam platform dengan skor rata-rata sebesar 4,55 atau setara dengan 90,91%.

3.2 Pembahasan

Keberhasilan penerapan metode SVD dalam pengembangan sistem rekomendasi produk di PT. Renos Marketplace Indonesia tercermin dari capaian evaluasi metrik akurasi seperti RMSE sebesar 0,1886; MSE sebesar 0,0366; dan MAE sebesar 0,1242. Nilai-nilai tersebut menunjukkan tingkat kesalahan prediksi yang rendah, mengindikasikan bahwa sistem mampu menyajikan rekomendasi yang mendekati preferensi aktual pengguna. Hal ini memperkuat temuan dari Sitorus yang juga menunjukkan bahwa algoritma SVD dapat mencapai RMSE rendah dan MAE minimal pada sistem rekomendasi film, menunjukkan konsistensi efektivitas SVD lintas domain [9]. Dengan demikian, hasil ini menegaskan bahwa SVD merupakan pendekatan yang tepat dalam menghadirkan rekomendasi berbasis data historis pengguna secara akurat.

Selain itu, hasil survei pengguna memperkuat efektivitas sistem dari sisi persepsi langsung. Seluruh indikator dalam survei menunjukkan skor di atas 80%, yang menandakan tingkat kepuasan pengguna berada dalam kategori sangat baik. Terutama, skor tertinggi pada pernyataan "Sistem rekomendasi ini dapat meningkatkan peluang saya untuk membeli produk di *platform* ini" menandakan bahwa pengguna tidak hanya merasa terbantu, tetapi juga ter dorong secara nyata untuk bertransaksi. Ini sejalan dengan hasil penelitian Hong et al. [12] yang membuktikan bahwa varian sederhana dari algoritma SVD mampu menghasilkan sistem yang efisien sekaligus tangguh terhadap *noise*, serta meningkatkan keterlibatan pengguna.

Sistem ini juga berhasil memenuhi tujuan utama dari penelitian, yaitu meningkatkan relevansi dan personalisasi dalam proses rekomendasi produk. Data rekomendasi yang dihasilkan tidak lagi bersifat statis atau acak, tetapi mampu menyesuaikan diri dengan pola perilaku pengguna seperti histori belanja, produk favorit, hingga intensitas interaksi. Perubahan pendekatan dari sistem konvensional ke berbasis SVD telah memberikan dampak signifikan dalam pengalaman pengguna yang lebih kontekstual dan tepat sasaran.

Lebih lanjut, efektivitas sistem juga didukung oleh arsitektur *backend* yang memadai, seperti penggunaan VPS dengan pemisahan antara *staging* dan *production environment*. Infrastruktur ini memungkinkan pengolahan data berskala besar dan eksekusi algoritma secara berkala dengan kendali sumber daya yang efisien. Pendekatan ini sangat cocok untuk sistem rekomendasi berbasis model seperti SVD yang membutuhkan pembaruan berkala, tetapi

tidak harus berjalan secara *real-time*. Dengan konfigurasi yang terukur, sistem dapat tetap responsif dan terintegrasi dengan baik pada proses bisnis *platform* renos.id.

Keunggulan lain dari implementasi ini adalah fleksibilitas dalam pengaturan bobot parameter yang memengaruhi hasil rekomendasi, seperti pembelian, penilaian, favorit, dan produk dilihat. Penyesuaian parameter ini memungkinkan sistem untuk disesuaikan dengan prioritas strategi bisnis perusahaan. Selain meningkatkan kualitas hasil, fitur ini juga menambah nilai adaptif dari sistem terhadap dinamika pasar dan perilaku konsumen. Pengujian *black-box* menunjukkan bahwa fitur ini berjalan sesuai ekspektasi, dengan validasi *input* yang ketat serta respons sistem yang konsisten.

Meskipun hasil yang diperoleh menunjukkan performa tinggi, penelitian ini juga menemukan beberapa keterbatasan yang patut diperhatikan. Salah satunya adalah model belum berjalan secara *real-time*, melainkan diatur dalam interval waktu tertentu. Hal ini berpengaruh pada kemampuan sistem dalam menanggapi perubahan preferensi pengguna secara cepat. Selain itu, data yang digunakan masih bersifat *dummy*, sehingga potensi bias atau ketidaksesuaian dengan perilaku pengguna nyata tetap harus diuji lebih lanjut dengan data produksi.

Secara keseluruhan, penelitian ini memberikan kontribusi praktis dan teoretis dalam bidang sistem rekomendasi, khususnya pada integrasi SVD dalam arsitektur *backend e-commerce*. Hasil pengujian, evaluasi performa algoritma, serta umpan balik pengguna membuktikan bahwa pendekatan ini efektif dalam menghadirkan rekomendasi yang relevan, efisien, dan disukai pengguna. Dibandingkan dengan metode lain, SVD terbukti memiliki keseimbangan antara akurasi dan efisiensi komputasi. Temuan ini sejalan dan memperkuat hasil-hasil penelitian sebelumnya, sekaligus menjadi pijakan untuk pengembangan sistem rekomendasi yang lebih adaptif di masa depan.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem *backend* rekomendasi produk berbasis *Collaborative Filtering* menggunakan algoritma SVD di PT. Renos Marketplace Indonesia. Sistem ini mampu menghasilkan rekomendasi yang lebih personal dan relevan dengan preferensi pengguna melalui integrasi berbagai parameter interaksi seperti pembelian, penilaian, favorit, dan produk dilihat. Evaluasi performa menunjukkan hasil akurasi yang tinggi dengan nilai RMSE, MSE, dan MAE yang rendah, serta mendapat respons positif dari pengguna internal melalui survei Skala *Likert*, di mana seluruh indikator memperoleh skor di atas 80%. Temuan ini menegaskan bahwa pendekatan SVD efektif dalam mendukung sistem rekomendasi yang responsif, fleksibel, dan sesuai dengan tujuan personalisasi pengalaman belanja pengguna.

Untuk meningkatkan efektivitas sistem ke depannya, disarankan agar pengembangan mencakup penambahan parameter perilaku pengguna seperti *clickstream*, waktu kunjungan, data pencarian pengguna dan data demografis guna memperkaya konteks rekomendasi. Sistem juga dapat dieksplorasi menggunakan pendekatan berbasis *deep learning* untuk personalisasi lebih mendalam [22]. *Monitoring* performa secara berkala perlu dilakukan agar bobot parameter tetap relevan terhadap dinamika pengguna. Selain itu, infrastruktur data harus terus ditingkatkan agar sistem tetap stabil dalam menangani volume data besar. Terakhir, integrasi dengan *platform* analitik seperti Twilio Segment dapat dipertimbangkan untuk pemetaan perilaku *real-time* dan optimalisasi segmentasi pengguna.

DAFTAR PUSTAKA

- [1] M. R. S. A. Ubaidillah, M. J. F. Shalah, and M. Yasin, “UMKM dan *E-Commerce* Pada Teknologi,” *MENAWAN: Jurnal Riset dan Publikasi Ilmu Ekonomi*, vol. 2, no. 5, pp. 07–11, 2024, doi: 10.61132/menawan.v2i5.757.
- [2] C. S. Octiva, P. E. Haes, T. I. Fajri, H. Eldo, and M. L. Hakim, “Implementasi Teknologi Informasi pada UMKM: Tantangan dan Peluang,” *Jurnal Minfo Polgan*, vol. 13, no. 1, pp. 815–821, 2024, doi: 10.33395/jmp.v13i1.13823.
- [3] S. Sarwindah, L. Laurentinus, O. Rizan, and H. Hamidah, “Memanfaatkan *Digital Marketing* bagi Usaha Rumahan Sayuran Hidroponik dengan *E-Commerce* sebagai Media Promosi,” *Jurnal Teknologi Terpadu*, vol. 7, no. 2, pp. 65–69, 2021, doi: 10.54914/jtt.v7i2.338.
- [4] S. Dewinta, “Pertumbuhan *E-Commerce* Makin Pesat, Asia Tenggara Semakin Berkilau.” Accessed: Feb. 21, 2025. [Online]. Available: <https://uwrite.id/news/pertumbuhan-e-commerce-makin-pesat-asia-tenggara-semakin-berkilau>
- [5] Statista, “Number of users in the eCommerce market in Indonesia from 2017 to 2029.” Accessed: Feb. 20, 2025. [Online]. Available: <https://www.statista.com/forecasts/251635/e-commerce-users-in-indonesia>.
- [6] Renos.id, “Mengenal Renos lebih jauh.” Accessed: Jul. 2, 2025. [Online]. Available: <https://www.renos.id/about/>
- [7] O. Lestari, P. Purwatiningsih, and R. S. Rini, “Analisis Pengukuran Kepuasan Konsumen dengan Pendekatan *Customer Satisfaction Indeks* pada *E-Commerce* di Jakarta,” *Jurnal Manajemen Kreatif dan Inovasi*, vol. 2, no. 3, pp. 180–191, 2024, doi: 10.59581/jmki-widyakarya.v2i3.3843.

- [8] C. Y. Hazizah, and T. Widyaningtyas, "Analisis Metode Collaborative Filtering menggunakan KNN dan SVD ++ untuk Rekomendasi Produk E-commerce Tokopedia," *Edumatic : Jurnal Pendidikan Informatika*, vol. 8, no. 2, pp. 595–604, 2024, doi: <https://doi.org/10.29408/edumatic.v8i2.27793>.
- [9] M. R. Sitorus, "Implementasi Model-Based Collaborative Filtering pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (*Singular Value Decomposition*)," Skripsi, Fakultas Teknik, Universitas Medan, Medan, Indonesia, 2023.
- [10] X. Ding, W. Yu, Y. Xie, and S. Liu, "Efficient Model-Based Collaborative Filtering with Fast Adaptive PCA," *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, vol. 2020-Novem, pp. 955–960, 2020, doi: [10.1109/ICTAI50040.2020.00149](https://doi.org/10.1109/ICTAI50040.2020.00149).
- [11] M. Kabić, G. D. López, and D. Keller, "A Refined SVD Algorithm for Collaborative Filtering," pp. 1–4, Dec. 2020, doi: [10.48550/arXiv.2012.06923](https://arxiv.org/abs/2012.06923).
- [12] S. Hong, J. Choi, Y. C. Lee, S. Kumar, and N. Park, "SVD-AE: Simple Autoencoders for Collaborative Filtering," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, {IJCAI-24}*, K. Larson, Ed., International Joint Conferences on Artificial Intelligence Organization, 2024, pp. 2054–2062. doi: [10.24963/ijcai.2024/227](https://doi.org/10.24963/ijcai.2024/227).
- [13] G. P. Insany, S. Somantri, and P. P. Amalia, "Implementasi Sistem Rekomendasi dengan Content Based Filtering dan Teknologi Virtual Tour Untuk Strategi Pemasaran Pada Website," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 1, pp. 300–313, 2024, doi: [10.47065/bits.v6i1.5358](https://doi.org/10.47065/bits.v6i1.5358).
- [14] F. K. Kartono *et al.*, "Pengujian Black Box Testing pada Sistem Website Osha Snack: Pendekatan Teknik Boundary Value Analysis," *Jurnal Kridatama Sains Dan Teknologi*, vol. 6, no. 02, pp. 754–766, 2024, doi: [10.53863/kst.v6i02.1407](https://doi.org/10.53863/kst.v6i02.1407).
- [15] I. R. Suci, N. A. Prasetyo, and G. F. Fitriana, "Buku Tamu Perpustakan Berbasis Website dengan Metode Agile (Perpustakaan Badan Pusat Statistik Kabupaten Brebes)," *Jurnal Teknologi Terpadu*, vol. 7, no. 2, pp. 70–76, 2021, doi: [10.54914/jtt.v7i2.390](https://doi.org/10.54914/jtt.v7i2.390).
- [16] A. Fareed, S. Hassan, S. B. Belhaouari, and Z. Halim, "A Collaborative Filtering Recommendation Framework Utilizing Social Networks," *Machine Learning with Applications*, vol. 14, no. January, p. 100495, 2023, doi: [10.1016/j.mlwa.2023.100495](https://doi.org/10.1016/j.mlwa.2023.100495).
- [17] A. Sitanggang, "Sistem Rekomendasi Anime Menggunakan Metode Singular Value Decomposition (SVD) dan Cosine Similarity," *Jurnal Teknologi Informasi*, vol. 2, no. 2, p. 90, 2023, doi: [10.35308/jti.v2i2.7787](https://doi.org/10.35308/jti.v2i2.7787).
- [18] X. Zhou, Y. Cui, N. Ma, X. Liu, L. Li, and L. Wang, "Influence of Characteristic Parameters of Signal on Fault Feature Extraction of Singular Value Method," *Journal of Vibroengineering*, vol. 22, no. 3, pp. 536–555, 2020, doi: [10.21595/jve.2019.20735](https://doi.org/10.21595/jve.2019.20735).
- [19] M. Loukili and F. Messaoudi, "Collaborative Singular Value Decomposition With User-Item Interaction Expansion For First-Time User and Item Recommendations," *International Journal of Informatics and Communication Technology*, vol. 14, no. 1, pp. 111–121, 2025, doi: [10.11591/ijict.v14i1.pp111-121](https://doi.org/10.11591/ijict.v14i1.pp111-121).
- [20] H. Hartatik and R. Rosyid, "Pengaruh User Profiling pada Rekomendasi Sistem Menggunakan K Means dan KNN," *Journal of Information System Management (JOISM)*, vol. 2, no. 1, pp. 13–18, 2020, doi: [10.24076/joism.2020v2i1.199](https://doi.org/10.24076/joism.2020v2i1.199).
- [21] M. Kuanr and P. Mohapatra, "Assessment Methods for Evaluation of Recommender Systems: A Survey," *Foundations of Computing and Decision Sciences*, vol. 46, no. 4, pp. 393–421, 2021, doi: [10.2478/fcds-2021-0023](https://doi.org/10.2478/fcds-2021-0023).
- [22] D. P. Romadhon and R. E. Putra, "Penerapan Metode Deep Learning Menggunakan Algoritma CNN based Recommendation pada Aplikasi E-Commerce Gols (Studi Kasus : PT. Cipta Giri Sentosa)," *Journal of Informatics and Computer Science (JINACS)*, vol. 5, no. 04, pp. 616–627, 2024, doi: [10.26740/jinacs.v5n04.p616-627](https://doi.org/10.26740/jinacs.v5n04.p616-627).