

Sistem Rekomendasi Suku Cadang Berdasarkan Item Based Filtering

<http://dx.doi.org/10.28932/jutisi.v7i1.3036>

Riwayat Artikel

Received: 12 Oktober 2020 | Final Revision: 22 Maret 2021 | Accepted: 24 Maret 2021

Christian Wibisono ^{#1}, Lucky Surya Haryadi ^{#2}, Juan Elisha Widyaya, ^{#3} Swat Lie Liliawati ^{#4}

^{#1}PT. ATEJA Tritunggal

Jl. Raya Batujajar KM 2.8, Bandung

¹christ.heaven22@gmail.com

^{#2}PT. Wijaya Adiputra Sentosa

Jl. Jendral Sudirman No. 257, Bandung

²suryaharyadi@gmail.com

^{#3}PT. Yamaha Indonesia Motor Mfg.

Jl. Soekarno Hatta 474a, Bandung

³juan.scholarium@gmail.com

^{#4}Magister Ilmu Komputer Universitas Kristen Maranatha

Jl. Prof. drg. Surya Sumantri No. 65, Bandung

⁴swat.11@it.maranatha.edu

Abstract — Replaceable spare parts in the workshop have many transactions and possibility thus a recommender system is needed to simplify the selection process. We propose a recommender system with item collaborative filtering, with high data sparsity. With Singular Value Decomposition (SVD) we reduce the matrix to improve the system and decrease “noise” value. The model will be evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Fraction of Concordant Pairs (FCP) metrics. The results of the recommendation model are MAE = 1.2752, RMSE = 1.4882, and FCP = 0.4947.

Keywords— collaborative filtering, recommender system, Single Value Decomposition, spare part

I. PENDAHULUAN

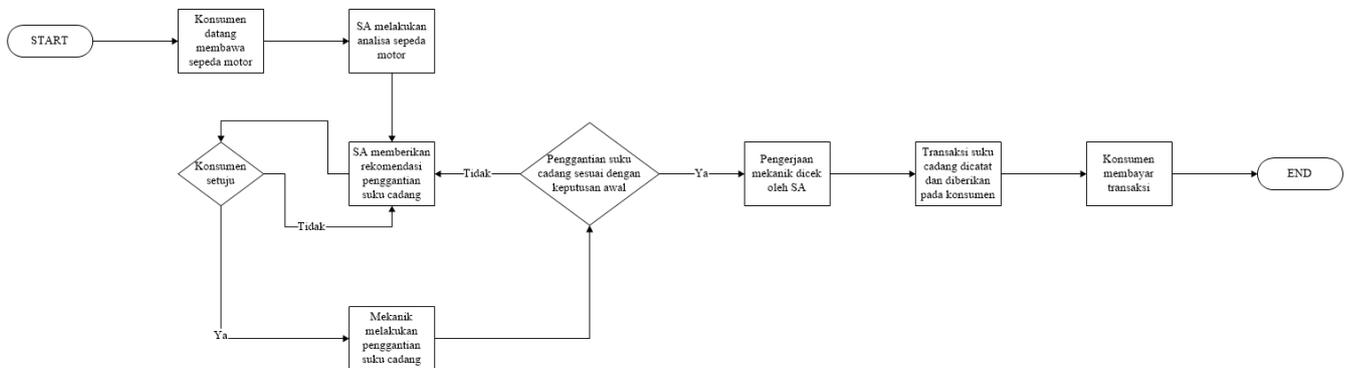
Dalam perawatan sepeda motor diperlukan ketepatan analisa pengerjaan dan suku cadang yang sesuai untuk memperbaiki motor. Satu sepeda motor terdiri dari banyak suku cadang dan terdapat ratusan suku cadang yang dapat diganti. Jenis suku cadang yang perlu diganti dan perawatan apa saja yang perlu dilakukan menjadi tanggung jawab seorang *service advisor* (SA). Pada Gambar 1 bisa dilihat alur perawatan sepeda motor ketika konsumen masuk ke dalam bengkel. Saat konsumen tiba di bengkel, SA akan memeriksa kondisi motor konsumen. Berdasarkan hasil pemeriksaan

tersebut, SA akan menyarankan perawatan dan penggantian suku cadang yang perlu dilakukan. Sesuai persetujuan konsumen, SA akan meminta salah satu mekanik untuk mengganti suku cadang pada motor yang bersangkutan dan melakukan perawatan sepeda motor. Jika saat pengerjaan ditemukan bagian lain yang sekiranya perlu dilakukan penggantian suku cadang, maka teknisi akan memberitahu SA agar ditanyakan kembali kepada konsumen mengenai suku cadang lainnya yang harus diganti. Setelah motor selesai dikerjakan, konsumen tinggal membayar biaya penggantian suku cadang dan konsumen bisa membawa motornya pulang.

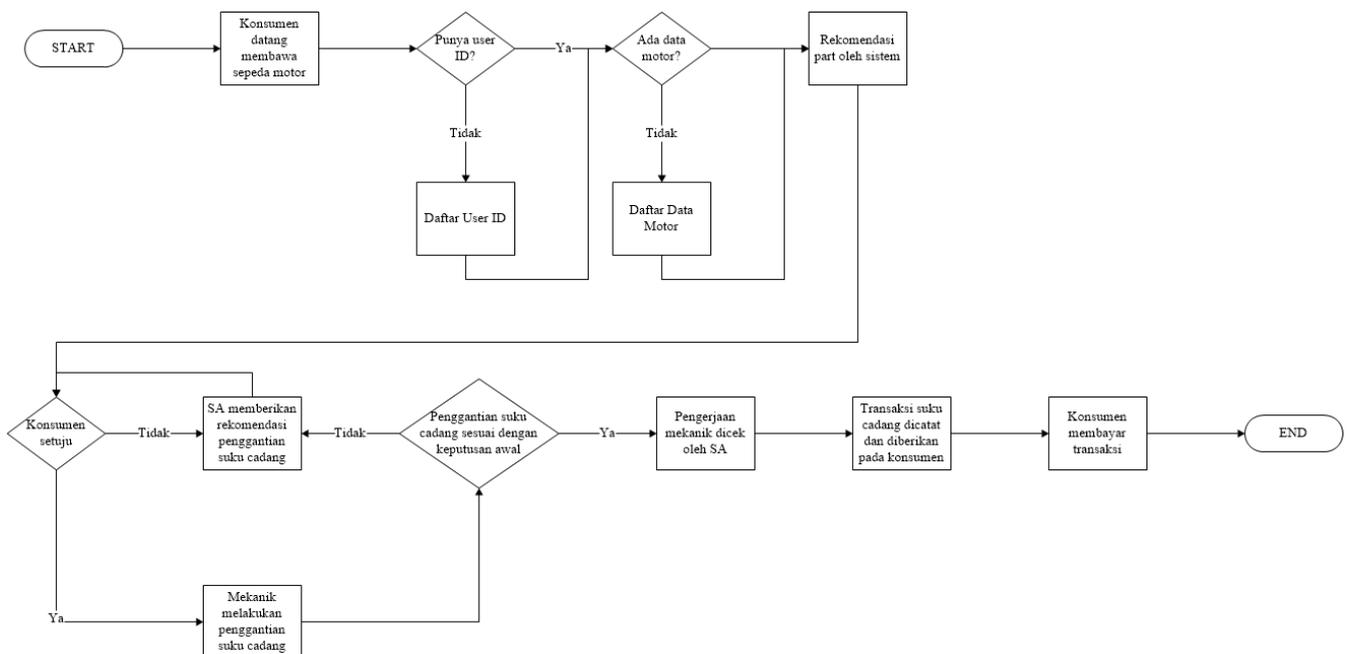
Rekomendasi pemilihan suku cadang biasanya mengacu pada buku perawatan yang diberikan oleh pabrik motor. Namun pada kenyataannya, terdapat perbedaan kebutuhan suku cadang antara kondisi aktual pada motor dengan yang disarankan oleh buku petunjuk perawatan. Faktor seperti cara pemakaian, cuaca, ataupun kondisi jalan menyebabkan kebutuhan suku cadang dan perbaikan setiap konsumen berbeda-beda. SA akan merekomendasikan suku cadang dengan memanfaatkan data konsumen dan sejarah belanja yang dipadukan dengan kondisi sepeda motor untuk meningkatkan pengalaman konsumen di bengkel. Jenis perawatan sepeda motor dan kebutuhan pergantian suku cadang menjadi beragam satu sama lain sehingga dibutuhkan rekomendasi pergantian suku cadang. Sistem rekomendasi dapat mengambil peran sebagai alat bantu bagi konsumen dan SA dalam memilih suku cadang yang tepat.

Pada Gambar 2 dapat dilihat alur perawatan sepeda motor ketika konsumen masuk ke dalam bengkel yang sudah menggunakan sistem rekomendasi suku cadang. Ketika konsumen datang ke bengkel, konsumen akan mengisi terlebih dahulu *user ID* dan data sepeda motor. Dari data-data ini akan dibangun rekomendasi suku cadang yang dikonfirmasi ulang kepada konsumen untuk meminta

persetujuan penggantian suku cadang. Jika konsumen tidak setuju, SA akan menyesuaikan ulang jenis suku cadang yang disarankan kepada konsumen sesuai dengan kondisi aktual pada sepeda motor. Setelah konsumen setuju atas suku cadang yang akan diganti, maka perawatan sepeda motor akan segera dilakukan oleh teknisi.



Gambar 1. Alur kerja konsumen ketika datang ke bengkel



Gambar 2. Alur kerja bengkel dengan sistem rekomendasi

Sistem rekomendasi banyak digunakan pada sistem *e-commerce* seperti Netflix [1] Amazon hingga pertemanan di Facebook atau Instagram dan sudah mulai merambah ke dunia *offline* seperti yang dilakukan Alibaba pada supermarket Hema. Penggunaan sistem informasi dengan pelayanan *offline* membuat paradigma baru bahwa perusahaan retail bisa bertindak sebagai *smart store* [2]. Dengan adanya sistem rekomendasi pada bengkel diharapkan bisa membantu kinerja SA dalam memilih suku cadang untuk

ditawarkan ke konsumen, sedangkan bagi konsumen juga mendapatkan kemudahan untuk memilih suku cadang yang sesuai dan tentu saja dari sisi bisnis diharapkan bisa meningkatkan penjualan suku cadang.

Penelitian ini akan berfokus pada penerapan algoritma SVD untuk membangun sistem rekomendasi suku cadang di bengkel. Sehingga pertanyaan yang akan dijawab pada penelitian ini adalah: bagaimana membangun sistem rekomendasi suku cadang pada sebuah bengkel motor?

Adapun tujuan penelitian ini mengimplementasikan SVD untuk membangun sistem rekomendasi suku cadang pada sebuah bengkel.

Sumber data yang dibutuhkan untuk membangun sistem rekomendasi ini berasal dari catatan servis di sebuah bengkel sepeda motor. Data tersebut merupakan data transaksi saja, tanpa adanya data *rating*. Model dibangun menggunakan bahasa Python dengan *library* *numpy* dan *scikit-surprise*. Karena *library* yang digunakan membutuhkan data *rating* untuk menghitung rekomendasi, maka data *rating* dibangun secara acak dengan rentang nilai dari 1 sampai 5. Sehingga perlu diketahui bahwa data *rating* adalah data buatan.

Pada bagian satu ini sudah dibahas latar belakang masalah dan peran rekomendasi sistem pada alur kerja bengkel yang diharapkan bisa meningkatkan kinerja bengkel. Pada bagian dua akan dibahas dari sisi kajian literatur sistem rekomendasi itu sendiri, faktorisasi matriks SVD, dan metrik pengukuran yang digunakan. Selanjutnya pada bagian tiga akan dibahas dataset yang digunakan, model yang dikembangkan dan juga hasil evaluasi model. Penelitian ini akan ditutup dengan kesimpulan dan saran pada bagian empat.

II. KAJIAN LITERATUR

A. Rekomendasi Sistem

Secara garis besar pendekatan sistem rekomendasi dapat dibagi ke dalam dua pendekatan, yaitu pendekatan *content based* dan *collaborative filtering* [3]. Pendekatan *content based* mengukur keterkaitan karakteristik *item-item* yang memiliki kesamaan konten. *Content based* [4] biasanya digunakan dalam mencari kedekatan antar dokumen seperti pada rekomendasi berita. Algoritma yang digunakan beragam dari *ruang vektor model* (VSM) dengan *term frequency-inverse document frequency* (TF-IDF) hingga model probabilistik dengan *naive bayes* atau *decision tree*. *Content based* tidak membutuhkan profil dari *user* sehingga tidak memiliki masalah *cold-start*[5].

Collaborative filtering (CF) [4] adalah sebuah model rekomendasi yang menggunakan kedekatan antara *user* atau *item* dari pengumpulan data interaksi antara *user* dan *item*. Interaksi *user-item* yang dibangun akan membentuk sebuah matriks *user-item*. Dari matriks ini persamaan antara *user* atau *item* dikalkulasi dan dicocokkan, sehingga nilai yang menggambarkan kedekatan antar *user* atau *item* akan menjadi dasar rekomendasi. Hal ini dikenal dengan istilah *neighbour*. Pengukuran similaritas dengan *item-item* lebih umum digunakan karena lebih efektif. Seperti pada penelitian Sarwar dan kawan-kawan [6] menguji efektivitas sistem rekomendasi dengan teknik *memory-based item-item CF* yang mengukur kedekatan dengan *adjusted cosine similarity*. Amazon juga diketahui menggunakan teknik *item-item* seperti yang dilaporkan Linden dan kawan-kawan [7] mampu diuji pada data skala besar. Pengukuran kedekatan antar *item* dapat menggunakan model VSM seperti pada *cosine similarity* atau metode statistik dengan metode *pearson*

coefficient. Teknik rekomendasi CF dapat dibagi menjadi dua yaitu *memory-based* dan *model-based*.

CF berbasis memori dapat dicapai dengan dua cara melalui teknik *user-based* atau *item-based*. Teknik CF *user-based* menghitung kesamaan antara *user* dengan membandingkan peringkat mereka pada *item* yang sama, dan kemudian menghitung peringkat prediksi untuk suatu *item* oleh *user* aktif sebagai rata-rata tertimbang dari peringkat *item* oleh *user* yang mirip dengan *user* aktif di mana bobot adalah kesamaan dari *user* ini dengan *item* target.

Teknik CF *item-based* menghitung prediksi menggunakan kesamaan antara *item* dan bukan kesamaan antara *user*. CF *item-based* membangun model kesamaan *item* dengan mengambil semua *item* yang dinilai oleh *user* aktif dari matriks *user-item*, lalu menentukan seberapa mirip *item* yang diambil dengan *item* target. Sejumlah *k* *item* dapat dipilih sebagai batasan jumlah *item* yang paling mirip. Prediksi dibuat dengan mengambil rata-rata tertimbang dari nilai *user* aktif pada *item* serupa *k*. Beberapa jenis ukuran kesamaan dapat digunakan untuk menghitung kesamaan antara *user* atau *item*. Dua ukuran kesamaan yang paling populer adalah berbasis korelasi seperti *pearson coefficient* dan *cosine similarity*. Contoh perhitungan *cosine similarity* untuk mengukur kedekatan antara *item* A dan B adalah sebagai berikut:

$$sim(A, B) = \frac{n(A \cap B)}{\sqrt{n(A) \times n(B)}} \quad (1)$$

A dan B adalah *item-item* yang dipilih oleh *user* dan irisan A dan B adalah *item* yang dibeli baik oleh A dan B. Berikut adalah ilustrasi penilaian kedekatan dengan *cosine similarity*.

TABEL I
MARIKS USER-ITEM

	Item A	Item B	Item C	Item D
User A	0	1	1	0
User B	1	0	0	0
User C	0	1	1	0
User D	0	1	1	1
User E	1	0	1	0
Total	2	3	4	1

Dalam tabel ini diambil contoh kedekatan antara *Item B* dan *Item C*:

$$sim(B, C) = \frac{3}{\sqrt{4 \times 3}} = 0.8660$$

Prediksi dinilai ketika seorang *user* telah memilih *Item B*. Nilai kedekatan akan dibuat peringkat berdasarkan kedekatan antar *item*:

$$pred(a,p) = r_a * sim(i,p) \quad (2)$$

TABEL II
PERHITUNGAN PREDIKSI ITEM

	pred(B,n)
Item B	1
Item C	0.8660254038
Item D	0.5773502692
Item A	0

Kelemahan pada CF yang coba diatasi yaitu [5]:

- Cold-start. Ini merujuk pada situasi di mana pemberi rekomendasi tidak memiliki informasi yang memadai tentang seorang *user* atau *item* untuk membuat prediksi yang relevan. Ini adalah salah satu masalah utama yang mengurangi kinerja sistem rekomendasi. Salah satu cara teknik yang bisa digunakan untuk mengatasi masalah ini adalah dengan rekomendasi berdasarkan *item* terpopuler.
- Data sparsity. Ini adalah masalah yang terjadi sebagai akibat dari kurangnya informasi yang cukup, yaitu, ketika hanya beberapa dari total jumlah *item* yang tersedia dalam database dinilai oleh *user*. Ini selalu mengarah ke matriks *user-item* yang jarang, ketidakmampuan untuk menemukan tetangga yang terdekat dan akhirnya, menghasilkan rekomendasi yang lemah. Selain itu, data *sparsity* selalu mengarah ke masalah cakupan, yang merupakan persentase *item* dalam sistem yang dapat dibuat rekomendasi. SVD salah satu algoritma yang membantu beberapa masalah data *sparsity*.

B. Matriks Faktorisasi SVD

SVD adalah teknik faktorisasi matriks yang digunakan untuk menghasilkan faktor matriks yang lebih kecil [3]. Matriks M didefinisikan dengan $m \times n$, dengan k sebagai matriks reduksi didefinisikan dengan rumus:

$$M = U \Sigma V^T \quad (3)$$

Di mana:

- U adalah matriks ukuran $m \times r$ ortogonal matriks dari eigenvalues $M^T M$
- Σ adalah matriks ukuran $r \times r$ merupakan matriks diagonal yang memiliki nilai singular.
- V^T adalah matriks ukuran $r \times n$ ortogonal matriks dari eigenvalues MM^T

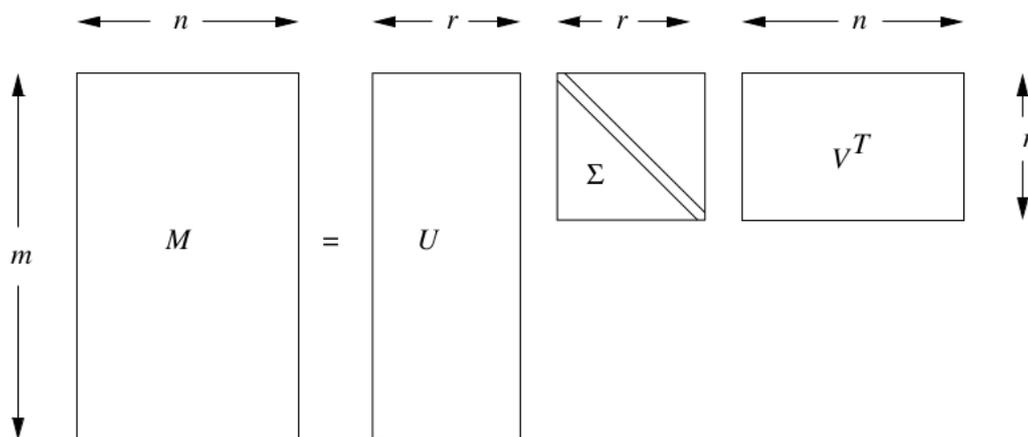
Matriks dapat direduksi dengan nilai pangkat k yang dapat membuat nilai matriks Σ menyesuaikan dengan pangkat k dan diubah menjadi nilai tiga matriks faktorisasi yaitu $m \times k$, $k \times k$, dan $k \times n$, dimana $k < r$. SVD menyediakan peringkat k rendah yang tetap mempertahankan dari nilai asli M dengan memperkecil nilai singular k . Disebut dengan Σ_k reduksi matriks karena Σ dilakukan dengan mempertahankan nilai Σ_k . Matriks U dan V juga direduksi menghasilkan matriks U_k dan V_k . Matriks U_k diperoleh dengan menghapus $(r - k)$ kolom matriks U dan matriks V_k diperoleh dengan menghapus $(r - k)$ baris matriks V . Ketika ketiga matriks direduksi akan memperoleh matriks M_k .

$$M_k = U_k \Sigma_k V_k^T \quad (4)$$

Sarwar [9] menunjukkan bahwa matriks yang lebih rendah dari ruang awal lebih baik karena penyaringan dari nilai-nilai kecil yang merupakan "noise", dalam relasi *user-item*.

Sebagai contoh seperti yang bisa dilihat pada Gambar 3 [3], misalkan matriks M memiliki dimensi $m \times n$ dan ranking M adalah r . Ranking matriks adalah jumlah baris terbesar (atau kolom ekuivalen) yang dapat dipilih, yang tidak ada kombinasi linier bukan nol dari baris yang merupakan vektor semua-nol 0 (atau dengan kata lain satu set baris atau kolom tersebut independen). Sehingga matriks U , Σ , dan V dengan properti berikut dapat dicari

- U adalah matriks kolom-ortonormal $m \times r$; artinya, masing-masing kolomnya adalah vektor satuan dan perkalian titik dari dua kolom mana pun adalah 0
- V adalah matriks kolom-ortonormal $n \times r$. Perhatikan bahwa V selalu dalam bentuk yang dialihkan, sehingga baris V^T ortonormal.
- Σ adalah matriks diagonal; Artinya, semua elemen yang tidak berada pada diagonal utama adalah 0. Elemen-elemen Σ disebut nilai singular M .



Gambar 3. Bentuk SVD [3]

Gambar 4 memberikan matriks peringkat-2 yang mewakili peringkat film oleh *user*. Dalam contoh ini terdapat dua “konsep” yang mendasari film: fiksi ilmiah dan romansa. Semua anak laki-laki hanya menilai fiksi ilmiah, dan semua perempuan menilai romansa saja. Keberadaan dua konsep yang secara ketat dipatuhi inilah yang memberi matriks peringkat 2. Artinya, salah satu dari empat baris pertama dan satu dari tiga baris terakhir dapat dipilih dan diamati bahwa tidak ada jumlah linier bukan nol dari baris-baris ini, yaitu 0. Tetapi tiga baris independen tidak dapat dipilih. Misalnya, jika baris 1, 2, dan 7, dipilih maka tiga dikalikan dengan yang pertama dikurangi yang kedua, ditambah nol dikalikan ketujuh adalah 0. Pengamatan serupa dapat dibuat untuk bagian kolom. Salah satu dari tiga kolom pertama dan salah satu dari dua kolom terakhir dapat dipilih, dan kolom tersebut akan independen, tetapi tidak ada himpunan tiga kolom yang independen. Penguraian matriks *M* dari Gambar 4 menjadi *U*, Σ , dan *V*, dengan semua elemen terkoreksi menjadi dua digit signifikan, ditunjukkan pada Gambar 5. Karena pangkat *M*

adalah 2, maka $r = 2$ bisa digunakan dalam dekomposisi. Rating *user* pada Gambar 4. dapat diuraikan dengan SVD menjadi seperti pada Gambar 5.

	Joe	Jim	John	Jack	Jill	Jenny	Jane
Matrix	1	3	4	5	0	0	0
Alien	1	3	4	5	0	0	0
Star Wars	1	3	4	5	0	0	0
Casablanca	0	0	0	0	4	5	2
Titanic	0	0	0	0	4	5	2

Gambar 4. User dan Item dalam sistem rekomendasi film [3]

$$\begin{matrix}
 \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix} & = & \begin{bmatrix} .14 & 0 \\ .42 & 0 \\ .56 & 0 \\ .70 & 0 \\ 0 & .60 \\ 0 & .75 \\ 0 & .30 \end{bmatrix} & \begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix} & \begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix} \\
 M & & U & \Sigma & V^T
 \end{matrix}$$

Gambar 5. SVD untuk matriks M [3]

C. Metrik Pengukuran

Metrik pengukuran yang digunakan pada model yang akan dibuat adalah *mean absolute error* (MAE), *root mean square error* (RMSE), dan *fraction of concordant pairs* (FCP) [10]. MAE dan RMSE digunakan untuk menghitung nilai error dari model yang dibangun, semakin kecil nilai MAE dan RMSE maka semakin baik model yang dihasilkan. MAE dapat dihitung dengan persamaan:

$$MAE = \frac{1}{|\hat{R}|} \sum_{r_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}| \quad (5)$$

Di mana:

- r_{ui} adalah nilai aktual dari *user u* terhadap *item i*
- \hat{r}_{ui} adalah nilai prediksi dari *user u* terhadap *item i*
- \hat{R} adalah total data point yang diprediksi

RMSE dapat dihitung dengan persamaan:

$$RMSE = \sqrt{\frac{1}{|\hat{R}|} \sum_{r_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2} \quad (6)$$

Di mana:

- r_{ui} adalah nilai aktual dari *user u* terhadap *item i*
- \hat{r}_{ui} adalah nilai prediksi dari *user u* terhadap *item i*
- \hat{R} adalah total data point yang diprediksi

FCP akan menghitung nilai *rating* yang diprediksi secara tepat oleh model. FCP merupakan metrik berorientasi peringkat yang bebas dari dua permasalahan pada RMSE. Menurut Yehuda Koren [10], RMSE tidak dapat mengekspresikan skala peringkat yang bervariasi di antara *user* yang berbeda dan tidak dapat diterapkan dalam kasus di mana peringkat bersifat ordinal. Jika diberikan set uji \hat{R} , maka nilai pasangan *concordant* untuk *user u* ditentukan dengan menghitung *rating* yang secara tepat diprediksi oleh model yang diberikan oleh persamaan:

$$n_c^u = |\{(i, j) \mid \hat{r}_{ui} > \hat{r}_{uj} \text{ dan } r_{ui} > r_{uj}\}| \quad (7)$$

Di mana:

- i, j adalah *item i* dan *j*
- r_{ui} adalah nilai aktual dari *user u* terhadap *item i*
- \hat{r}_{ui} adalah nilai prediksi dari *user u* terhadap *item i*
- r_{uj} adalah nilai aktual dari *user u* terhadap *item j*
- \hat{r}_{uj} adalah nilai prediksi dari *user u* terhadap *item j*

Dan nilai *discordant* n_d^u dihitung dengan cara yang sama, yaitu:

$$n_d^u = |\{(i, j) \mid \hat{r}_{ui} > \hat{r}_{uj} \text{ dan } r_{ui} < r_{uj}\}| \quad (8)$$

Dengan menjumlahkan terhadap seluruh *user* $n_c = \sum_u n_c^u$ dan $n_d = \sum_u n_d^u$. Metrik kualitas yang digunakan untuk mengukur proporsi pasangan *item* yang diprediksi dengan tepat dijabarkan dengan FCP. Semakin tinggi nilai FCP maka semakin baik model yang dihasilkan.

$$FCP = \frac{n_c}{n_c + n_d} \quad (9)$$

Di mana:

- n_c adalah nilai *concordant* dari semua *user*
- n_d adalah nilai *discordant* dari semua *user*

D. Penelitian Terkait

Adapun penelitian-penelitian lain yang menjadi acuan dalam menyusun penelitian ini bisa dilihat sebagai berikut.

TABEL III
HASIL PENELITIAN DHEERAJ KUMAR BOKDE DAN KAWAN-KAWAN

<i>Item-Based Collaborative Filtering</i>				
<i>Similarity Measure</i>	MAE		RMSE	
	70% <i>training</i>	80% <i>training</i>	70% <i>training</i>	80% <i>training</i>
<i>Pearson Correlation Similarity</i>	0.842	0.828	1.08	1.061
<i>Euclidean Distance Similarity</i>	0.816	0.818	1.022	1.026
<i>Log Likelihood Similarity</i>	0.814	0.817	1.019	1.025
<i>Tanimoto Coefficient Similarity</i>	0.793	0.794	0.999	1.002

Dheeraj Kumar Bokde dan kawan-kawan [11] melakukan *collaborative filtering* menggunakan teknik reduksi dimensi pada sebuah *framework* bernama Mahout. Pada penelitian terkait digunakan metode SVD untuk mendapatkan rekomendasi dari dataset yang digunakan. Dataset yang digunakan dibuat sendiri dalam sebuah *framework* Mahout. Dalam penerapan SVD dilakukan pendekatan *stochastic* di mana akan dibuat Gaussian matriks yang selanjutnya dilakukan perhitungan kolom ortogonal sampai didapatkan

nilai eigen. Penerapan SVD yang dilakukan dikombinasikan dengan pengurangan dimensi matriks setiap dilakukan faktorisasi sehingga akan didapatkan ukuran matriks yang lebih kecil dan akan mempengaruhi kecepatan sistem rekomendasi yang dibuat dalam memberikan sebuah rekomendasi. Dheeraj Kumar Bokde dan kawan-kawan juga menggunakan beberapa *similarity measure* yang berbeda pada sistem rekomendasi SVD yang dibuat, hasil dari penelitian terkait bisa dilihat pada Tabel III.

Manolis Vozalis dan kawan-kawan [12] melakukan analisis teknik-teknik SVD untuk melakukan *collaborative filtering*, adapun dataset yang digunakan adalah MovieLens dataset dan Jester dataset. Berdasarkan penelitian yang dilakukan digunakan tiga pendekatan metode CF yaitu SVD-CF, PCA-CF, dan CA-CF. Ketiga metode ini nantinya akan dibandingkan dengan nilai reduksi matriks yang dihasilkan oleh kedua dataset. Hasil dari penelitian Manolis Vozalis menyatakan bahwa metode CA-CF memiliki nilai akurasi tertinggi untuk nilai reduksi matriks rendah yaitu sebesar 0.855, untuk metode PCA-CF memiliki nilai yang stabil terlepas dari jumlah reduksi matriks dan SVD-CF memiliki akurasi yang tinggi untuk nilai reduksi matriks yang semakin banyak yaitu sebesar 0.872.

Zeinab Sharifi dan kawan-kawan [13] melakukan pengembangan algoritma SVD dalam merancang sebuah sistem rekomendasi menggunakan dataset MovieLens sebagai

sumber data yang digunakan. Salah satu metode yang digunakan adalah MIDItemSVD di mana SVD yang dibangun memiliki perbedaan dalam cara melakukan reduksi matriks, reduksi matriks dilakukan secara paralel hingga didapat sebuah matriks yang nilainya stabil sebelum dilakukan proses perhitungan nilai rekomendasi dari setiap *item* yang ada. Penelitian yang dilakukan oleh Zeinab Sharifi memiliki akurasi sebesar 0.832.

III. MODEL REKOMENDASI SISTEM

A. Persiapan Dataset

Sumber data yang dibutuhkan untuk membangun sistem rekomendasi ini berasal dari catatan servis di sebuah bengkel sepeda motor. Data transaksi per bulan kira-kira ada 1.200 sampai 1.400 transaksi, per transaksi bisa terdapat empat sampai lima jenis suku cadang yang diganti. Data yang diambil adalah data transaksi selama enam bulan (Jan sampai Jun 2019) sebagai dataset yang digunakan. Model sepeda motor yang akan dipakai untuk eksperimen dan evaluasi sistem adalah model NM. Dalam rentang enam bulan tersebut didapati total transaksi model NM sebanyak 3.041 yang terdiri dari jumlah *user* 2027 dan jumlah *item* 367. Perpaduan *user* dan *item* menghasilkan ukuran matriks sebesar 743.909 dan tingkat *sparsity* data 99.03%.

USER_NAME	MODEL_CATEGORY	MILAGE	PART_NO	PART_TYPE
A1	NM	6,681	5YPF58Z5Z1ZZ	non captive
A1	NM	6,681	9Z793AY8Z3ZZ	Chemical (Non Engine Oil)
A1	NM	6,681	9Z798CZ184ZZ	Accessories
A1	NM	6,681	9Z793AJ425ZZ	Oil(4T)
A2	NM	3,292	9Z793AJ425ZZ	Oil(4T)
A3	NM	24,721	2DPF58Z5ZZZZ	non captive
A3	NM	24,721	2DPE4451ZZZZ	4 Cycle Matic
A3	NM	24,721	9Z793AJ8Z5ZZ	4 Cycle Matic
A3	NM	24,721	54PH189ZZ2ZZ	4 Cycle Matic
A3	NM	24,721	947ZZZZ415ZZ	non captive
A3	NM	24,721	2DPE7641ZZZZ	non captive
A3	NM	24,721	9Z793AJ829ZZ	Oil(4T)
A4	NM	4,607	9Z793AJ425ZZ	Oil(4T)

Gambar 6. Dataset yang sudah dibersihkan

PENGGUNA	GREASE C	LEVER 2'	VISOR'	Y Gear Oil 1	PROTECTO	SEAL FEL	LENS FLAS	'DAMPER L
A1	0	0	0	0	0	0	0	0
A2	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	0	0	0
A4	0	0	0	0	0	0	0	0
A5	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	0	0
A9	1	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0
A12	1	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0
A14	0	0	0	0	0	0	0	0

Gambar 7. Matriks user-item

Dataset tersebut mula-mula dibersihkan terlebih dahulu dari unsur-unsur seperti tanda kurung, data transaksi yang melakukan servis tanpa pergantian *item* duplikasi kode suku cadang yang berbeda pada *item* yang sama, ini diakibatkan perubahan kode produksi di pabrik namun mengacu *item* yang sama. Setelah data dibersihkan, maka matriks *user-item* bisa dibangun. Matriks ini merupakan pasangan antara *user* dengan *item* yang pernah dibelinya. Dataset yang sudah dibersihkan bisa dilihat pada Gambar 6 dan matriks *user-item* bisa dilihat pada Gambar 7.

Dataset merupakan data transaksi saja, tanpa adanya data *rating*. Karena *library* yang digunakan membutuhkan data *rating* untuk menghitung rekomendasi, maka data *rating* akan dibangun secara acak dengan rentang dari 1 sampai 5.

B. Membangun Model Rekomendasi

Model yang digunakan untuk memberikan rekomendasi 5 *item* menggunakan *numpy* yang merupakan salah satu *library* besar yang ada di dalam bahasa pemrograman Python. Namun sebelum melakukan pembentukan model SVD menggunakan *numpy* hal pertama yang perlu dilakukan adalah melakukan transformasi data yang semula berupa record array satu dimensi menjadi array dua dimensi atau matriks.

Untuk menyesuaikan dengan kebutuhan model SVD yang ada pada *numpy* maka matriks yang sudah ada perlu dilakukan sedikit penyesuaian dimana perlu dilakukan *masking id* untuk *id* unik dari *user* maupun *item* atau *part*. *Masking* disini bertujuan untuk memberikan identitas baru berupa nilai integer. Hal ini dikarenakan matriks yang ada hanya boleh memiliki nilai *integer* agar bisa diproses oleh pemodelan SVD yang ada pada *numpy*.

Sesudah proses *masking* dijalankan maka tahapan berikutnya adalah melakukan perubahan baris dan kolom pada matriks yang ada untuk menciptakan pemodelan. Row merupakan *id part* dan kolom merupakan *id user*, dimana value matriks yang ada hanya mengambil nilai *rating* saja. Untuk matriks bisa dilihat pada Gambar 8.

```
array([
[2,53,10,....,0,200,100],
[32,3,203,....,188,10,216],
[203,127,4,....,252,203,127],
...,
[120,67,2,....,2,2,3],
[172,215,203,....,202,172,215],
[203,127,0,....,215,203,127]], dtype = u int 8)
```

Gambar 8. Matriks part dan user

Sesudah matriks terbentuk maka tahapan selanjutnya adalah melakukan normalisasi matriks yang sudah ada. Matriks yang sudah dinormalisasi akan digunakan oleh fungsi SVD yang ada di dalam *library* *numpy*. *Output* yang dihasilkan akan terdiri dari 3 buah matriks yaitu matriks ortogonal, matriks diagonal dan matriks ortogonal transpose, dimana jika ketiga matriks tersebut dikalikan akan kembali menghasilkan matriks yang sudah dinormalisasi. Adapun kode program dalam bahasa Python dapat dilihat pada Algoritma 1.

Algoritma 1: Membangun SVD

Input: *normalised_mat*, *ratings_mat*

Output: *U*, *S*, *V*

```
A = normalized_mat.T /
    np.sqrt(ratings_mat.shape[0] - 1)
U, S, V = np.linalg.svd(A)
```

Tahapan selanjutnya adalah membuat sebuah fungsi yang menerima *input* berupa matriks hasil perkalian matriks ortogonal transpose dengan matriks transpose, *id part* yang

ingin dicari rekomendasinya, dan N yang mewakili nilai jumlah rekomendasi yang akan digunakan.

Fungsi yang dibuat akan melakukan pengecekan setiap *item*-nya menggunakan persamaan *similarity* yang hasilnya sementara akan disimpan dalam sebuah array *temporary*. Setelah semua proses selesai dijalankan maka akan diambil nilai sebanyak top N dari array yang sudah dibentuk. Sesudah fungsi ini dibuat maka menggunakan cukup memasukan *id part* dan nilai N rekomendasi untuk mendapatkan N *item* rekomendasi dari sistem seperti yang ditunjukkan oleh Gambar 9.

```
1 print_similar_parts(part_data, part_id, indexes)
Recommendations for BEARING 2DP1:
BEARING 2DP1
BODY COWLING 2 MNM3 BV31
BEARING 2PH1
BEARING 3C11
BEARING 5C81 2S62 5D21
```

Gambar 9. Hasil Rekomendasi

C. Evaluasi Model Rekomendasi

Untuk evaluasi model yang sudah dibuat diperlukan *library* tambahan yaitu *scikit-surprise*, *library* ini memiliki salah satu fungsi untuk mengukur tingkat akurasi dan kemungkinan apakah *item* tertentu bisa direkomendasikan atau tidak berdasarkan pemodelan SVD yang sudah dibuat sebelumnya. *Library* ini digunakan dikarenakan *library* *numpy* tidak menyediakan fitur pengecekan hasil rekomendasi dan akurasi yang dihasilkan oleh model.

Yang dilakukan pertama adalah melakukan pembuatan model akurasi, dimana model ini membutuhkan nilai dari masing-masing *user*, *part* dan *rating* namun tidak perlu dalam bentuk matriks. Sesudah itu dilakukan proses *splitting* sebanyak 80% untuk data *training* dan 20% untuk data test. Pada proses *training* ditambahkan pula faktor random sebanyak 50% untuk memastikan model SVD yang sudah dibentuk dapat digunakan pada *environment* yang berbeda.

Sesudah proses *training* selesai tahapan berikutnya adalah membuat model prediksi menggunakan model SVD dengan *library* *surprise*. Prediksi yang dihasilkan membutuhkan 2 parameter *input* yaitu *id user* dan *id part*, adapun *output* yang dihasilkan adalah nilai estimasi kecocokan rekomendasi dan detail yang memiliki informasi apakah *item* tersebut bisa direkomendasikan atau tidak beserta dengan alasannya seperti yang dapat dilihat pada Gambar 10.

```
1 recom_svd = algo_svd.predict(uid='1',iid='1')
2 recom_svd
Prediction(uid='1', iid='1', r_ui=None, est=2.9689575033200533, details={'was_impossible': False})
```

Gambar 10. Hasil prediksi model SVD

Tahapan akhir adalah melakukan uji akurasi dari model yang sudah ada, pada tahapan ini dipanggil beberapa fungsi dari *library* *scikit-surprise* untuk menghitung nilai MAE, RMSE dan FCP seperti yang dapat dilihat pada Tabel IV.

Model ini dibandingkan dengan performa MAEN, RMSE dan FCP dengan algoritma *k-nearest neighbor* (KNN) *baseline* sebagai perbandingan.

KNN adalah model dasar yang baik untuk perbandingan sistem pemberi rekomendasi. KNN adalah metode pembelajaran *lazy* non-parametrik. KNN menggunakan kumpulan data di mana titik data dipisahkan menjadi beberapa *cluster* untuk membuat inferensi untuk sampel baru. KNN tidak membuat asumsi apa pun pada distribusi data yang mendasarinya tetapi mengandalkan kesamaan fitur *item*.

TABEL IV
HASIL PENGUJIAN SVD DAN KNN BASELINE

Metrik	SVD	KNN
MAE	1.2752	1.2827
FCP	0.4947	0.4913
RMSE	1.4882	1.5165

Hasil penelitian menunjukkan bahwa algoritma SVD sedikit lebih baik jika dibandingkan dengan algoritma KNN ketika membandingkan kinerja MAE, RMSE, dan FCP. MAE, RMSE, dan FCP membandingkan antara nilai aktual dengan nilai prediksi. Semakin kecil nilai MAE dan nilai RMSE maka model semakin baik. Sedangkan semakin besar nilai FCP yang diberikan oleh model, maka model tersebut semakin baik. Keuntungan menggunakan RMSE dibandingkan MAE adalah bahwa kesalahan pada RMSE dihitung lebih besar dibandingkan MAE.

IV. SIMPULAN

Pada penelitian ini sistem rekomendasi dibangun untuk merekomendasikan suku cadang pada sebuah bengkel otomotif. Model rekomendasi dibangun dengan pendekatan matriks faktorisasi SVD untuk mengatasi masalah sparsity data. Hasil penelitian menunjukkan bahwa algoritma SVD sedikit lebih baik jika dibandingkan dengan algoritma KNN ketika membandingkan kinerja MAE, RMSE, dan FCP. Model SVD memberikan nilai MAE = 1.2752, RMSE = 1.4882, dan FCP = 0.4947. Semakin kecil nilai RMSE dan nilai MAE maka model semakin baik. Sedangkan semakin besar nilai FCP yang diberikan oleh model, maka model tersebut semakin baik.

Penelitian ini bisa dikembangkan dengan menggunakan dataset yang lebih akurat, yaitu menggunakan dataset yang memiliki nilai *rating* eksplisit dari user dengan rentang 1 sampai 5. Selain itu, model sistem rekomendasi dapat diimplementasikan secara aktual pada operasional sebuah bengkel untuk mengetahui manfaat model untuk bisnis sepeda motor.

DAFTAR PUSTAKA

- [1] Y. Koren, "The BellKor Solution to the Netflix Grand Prize," hlm. 10.
- [2] J. Hanke, M. Hauser, dan A. Dă, "Redefining the Offline Retail Experience: Designing Product Recommendation Systems for Fashion Stores," hlm. 15, 2018.
- [3] J. Leskovec, A. Rajaraman, dan J. D. Ullman, "Mining of Massive Datasets."
- [4] M. T. Jones, "Recommender systems, Part 1: Introduction to approaches and algorithms," *Recomm. Syst.*, hlm. 7.
- [5] F. O. Isinkaye, Y. O. Folajimi, dan B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egypt. Inform. J.*, vol. 16, no. 3, hlm. 261–273, Nov 2015, doi: 10.1016/j.eij.2015.06.005.
- [6] B. Sarwar, G. Karypis, J. Konstan, dan J. Reidl, "Item-based collaborative filtering recommendation algorithms," dalam *Proceedings of the tenth international conference on World Wide Web - WWW '01*, Hong Kong, Hong Kong, 2001, hlm. 285–295, doi: 10.1145/371920.372071 [Online]. Tersedia pada: <http://portal.acm.org/citation.cfm?doid=371920.372071>. [Diakses: 26-Feb-2020]
- [7] G. Linden, B. Smith, dan J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, hlm. 76–80, Jan 2003, doi: 10.1109/MIC.2003.1167344.
- [8] L. Dzumiroh dan R. Saptono, "Penerapan Metode Collaborative Filtering Menggunakan Rating Implisit pada Sistem Rekomendasi Pemilihan Film di Rental VCD," *J. Teknol. Inf. ITSsmart*, vol. 1, no. 2, hlm. 54, Mar 2016, doi: 10.20961/its.v1i2.590.
- [9] B. Sarwar, G. Karypis, J. Konstan, dan J. Riedl, "Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems," hlm. 6.
- [10] Y. Koren dan J. Sill, "Collaborative Filtering on Ordinal User Feedback," hlm. 5.
- [11] D. kumar Bokde, S. Girase, dan D. Mukhopadhyay, "An Item-Based Collaborative Filtering using Dimensionality Reduction Techniques on Mahout Framework," *ArXiv150306562 Cs*, Mar 2015 [Online]. Tersedia pada: <http://arxiv.org/abs/1503.06562>. [Diakses: 22-Mar-2021]
- [12] M. Vozalis, A. Markos, dan K. Margaritis, *Evaluation of standard SVD-based techniques for Collaborative Filtering*.
- [13] Z. Sharifi, M. Rezghi, dan M. Nasiri, "New algorithm for recommender systems based on singular value decomposition method," dalam *ICCKE 2013*, 2013, hlm. 86–91, doi: 10.1109/ICCKE.2013.6682799.