**Aplikasi Rekomendasi E-book Pada Perpustakaan Digital Menggunakan Deep Collaborative Filtering Berbasis Mobile Dan Web**

**Evan Owen Pasaribu1, Iman Fajar Lombu2, Martua Sinaga3, Ronsen Purba4, Darwin5**

STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789

3Jurusan Teknik Informatika, STMIK Mikroskil, Medan

e-mail: [**1171111067@students.mikroskil.ac.id**](mailto:1171111067@students.mikroskil.ac.id), [2**171111768@students.mikroskil.ac.id**](mailto:2171111768@students.mikroskil.ac.id),[3**171112356@students.mikroskil.ac.id**](mailto:3171112356@students.mikroskil.ac.id), [4**ronsen@mikroskil.ac.id**](mailto:4ronsen@mikroskil.ac.id), [5**darwin.ng@mikroskil.ac.id**](mailto:5darwin.ng@mikroskil.ac.id)

***Abstrak***

*Perpustakaan digital mengalami permasalahan sparsity matriks rating karena pemberian rating yang cenderung jarang diberikan oleh user. Banyaknya user yang hanya menilai sejumlah kecil item, membuat sistem rekomendasi collaborative filtering memberikan hasil rekomendasi dengan akurasi yang sangat rendah. Oleh karena itu, dalam Tugas Akhir ini digunakan metode Deep Collaborative Filtering (DCF) yang merupakan penggabungan pendekatan collaborative filtering yaitu probabilistic matrix factorization dengan salah satu teknik deep learning yaitu marginalized denoising auto-encoders untuk memprediksi rating user terhadap suatu item, sehingga dapat mengurangi sparsity matriks rating pada perpustakaan digital. Berdasarkan hasil pengujian dengan Black Box Testing perangkat lunak secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan. Hasil pengujian keakuratan deep collaborative filtering menunjukkan bahwa nilai error yang didapatkan menghasilkan nilai rata - rata error yang relatif rendah terhadap 30 user dan 100 item dengan 2847 jumlah data rating kosong yang diuji dengan nilai MAE 0,6347 dan RMSE 0,7514.*

**Kata Kunci :** *Sparsity*, DCF, *Collaborative filtering*, *Deep Learning.*

***Abstract***

*The digital library experienced the problem of sparsity a matrix rating because of giving its rating tends to be rarely given by users. The large number of anesthetizers who only assess a small number of items, making the collaborative filtering recommendation system provide a recommendation result with extremely low accuracy. Therefore, in the Final Task is used by the Deep Collaborative Filtering method (DCF) which is the merger of the collaborative filtering approach, namely the matrix factorization with deep learning techniques, namely auto-encoders to predict the rating of an item, so that it can reduce sparsity the matrix rating on the digital library. Based on the test results with the Black Box Software testing it functionally took out the results as expected. The results of the accuracy of deep collaborative filtering shows that the error value obtained produces the relatively low grades – the error average is relatively low to 30 users and 100 items with 2847 the number of empty rating data tested with MAE's value of 0.6599 and RMSE 0.753.*

**Keywords:** *Sparsity,* DCF*, Collaborative filtering, Deep Learning*.

## PENDAHULUAN

Saat ini sistem rekomendasi telah diterapkan di berbagai domain seperti musik, film, buku, dan produk[1]. Penggunaan teknik rekomendasi yang akurat dan efisien sangat penting bagi suatu sistem untuk memberikan hasil yang baik dan bermanfaat kepada setiap penggunanya[2]. Sistem rekomendasi *collaborative filtering* dapat merekomendasikan *item* berdasarkan kumpulan *user* dengan preferensi yang sama[3]. Tetapi, sistem rekomendasi ini mengalami masalah *sparsity* pada matriks *rating*. Banyak *user* hanya menilai sejumlah kecil *item* sehingga rekomendasi yang dihasilkan memiliki akurasi yang sangat rendah[4]. Sementara dalam konteks perpustakaan digital, banyaknya koleksi dan kebutuhan informasi yang beragam membuat *rating* dari *user* sangat diperlukan. Tetapi, *rating* pada *item* perpustakaan digital cenderung jarang diberikan oleh *user*[5].

*Deep learning*, sebagai jenis pendekatan *machine learning* telah berhasil diterapkan di banyak domain penelitian berbeda, seperti *computer vision*, *speech recognition*, *natural language processing* dan lain sebagainya[6]. Menggunakan *deep learning* pada sistem rekomendasi memungkinkan model secara otomatis mempelajari fitur yang digeneralisasi dengan baik bagi *user* dan *item* dari sumber daya yang berbeda, sehingga membuat sistem rekomendasi akanlebih memahami apa yang dibutuhkan *user* dan hal tersebut akan meningkatkan hasil rekomendasi[7]. Model *deep learning* untuk *collaborative filtering* merupakan penggabungan *collaborative filtering* berbasis matriks *factorization* dengan algoritma *deep learning*. *Deep collaborative filtering* yang merupakan model *hybrid*, menggunakan matriks *rating* dan *side information* serta mengintegrasikan matriks *factorization* dan fitur *learning* dengan memodelkan pemetaan antara *latent factors* yang digunakan dalam *collaborative filtering* dan *latent layers* dalam *deep models*[8].

Berdasarkan uraian di atas, maka dikembangkan sebuah aplikasi yang dapat memberikan rekomendasi kepada *user* dengan mengurangi *sparsity* pada matriks *rating* dan dituangkan dalam tugas akhir dengan judul “APLIKASI REKOMENDASI E-BOOK PADA PERPUSTAKAAN DIGITAL MENGGUNAKAN DEEP COLLABORATIVE FILTERING BERBASIS MOBILE DAN WEB”.

## METODE PENELITIAN

### Kajian Pustaka

#### Perpustakaan Digital

Kemajuan teknologi komunikasi informasi, internet, dan digitalisasi buku, surat kabar, majalah, jurnal, video ceramah, dan sebagian besar buku referensi mengubah konsep perpustakaan konvensional dengan model perpustakaan baru yang disebut *resource center*. Sumber daya digital ini disimpan menggunakan teknologi penyimpanan elektronik atau berlangganan dari penyedia layanan tersebut untuk mengurangi penyimpanan buku, jurnal, majalah berbasis kertas dan kemudahan dalam pencarian *database*, akses fasilitas, dan menekan biaya pemeliharaan perpustakaan[9].

Perpustakaan digital adalah bidang penelitian yang terus berkembang. Perpustakaan digital didefinisikan sebagai kumpulan informasi elektronik yang berisi *repository* objek digital yang besar dan beragam, dan dapat diakses oleh sejumlah besar *user* yang tersebar secara geografis. Objek digital yang dimaksud seperti teks, gambar, peta, suara, video, katalog, dan jurnal ilmiah[10]. Setiap *user* dapat mengakses sumber daya hingga informasi yang tersedia di perpustakaan dengan menggunakan komputer dan koneksi ke jaringan perpustakaan tanpa harus hadir secara fisik di perpustakaan[11].

#### Sistem Rekomendasi

Sistem rekomendasi merupakan fitur yang berfungsi untuk membantu *user* untuk menemukan saran tentang *item* apa yang sebaiknya digunakan atau dipilih. Secara umum sistem rekomendasi ditujukan untuk individu yang kekurangan pengalaman atau kompetensi yang cukup untuk mengevaluasi banyaknya jumlah alternatif *item* yang ada pada suatu kasus tertentu[12]. Saat ini sistem rekomendasi telah diterapkan di berbagai domain seperti musik, film, buku, dan produk. Fungsi tersebut tidak hanya dapat merekomendasikan produk yang disukai *user* tetapi juga dapat secara aktif mempromosikan pemasaran produk. Dasar dari sistem rekomendasi dibangun dengan asumsi bahwa riwayat *user* internet dapat sepenuhnya mewakili preferensi *user*, dan asumsi ini berarti bahwa preferensi *user* dapat diperoleh dengan *data mining* dan analisis riwayat *user*[1].

Penggunaan teknik rekomendasi yang akurat dan efisien sangat penting bagi suatu sistem untuk memberikan hasil yang baik dan bermanfaat kepada setiap penggunaannya. Salah satu masalah yang menyebabkan sistem rekomendasi tidak akurat adalah karena terjadinya *sparsity*. *Sparsity* merupakan masalah yang terjadi akibat kurangnya informasi yang cukup, yaitu hanya sedikit dari jumlah *item* yang tersedia dalam *database* yang dinilai oleh *user.* Akibatnya, *user* tidak mendapatkan hasil rekomendasi yang baik dan bermanfaat[2].

#### Deep Learning

*Deep learning* adalah bentuk *machine learning* yang memungkinkan komputer untuk belajar dari pengalaman dan memahami dunia dalam hal hierarki konsep. Karena komputer mengumpulkan pengetahuan dari pengalaman, sehingga operator komputer manusia tidak diperlukan untuk menentukan semua pengetahuan yang dibutuhkan oleh komputer. Hierarki konsep memungkinkan komputer untuk mempelajari konsep yang rumit dengan membangunnya menjadi konsep yang lebih sederhana. Grafik hierarki ini akan memiliki banyak lapisan[13].

*Deep learning* telah berhasil diterapkan di banyak domain penelitian berbeda, seperti *computer vision*, *speech recognition*, *natural language processing* dan lain sebagainya.Pada bidang ini, dibandingkan dengan pendekatan tradisional, pendekatan *deep learning* meningkatkan kinerja secara luar biasa. Karena keberhasilan besar *deep learning*, beberapa peneliti mencoba menggunakannya dalam sistem rekomendasi dan berharap dapat meningkatkan kinerja sistem rekomendasi, seperti halnya penerapan model *deep learning* di bidang penelitian lain[6]. Dalam satu dekade terakhir, banyak sekali penelitian yang mencoba untuk memperkenalkan metode *deep learning* ke dalam sistem rekomendasi untuk meningkatkan kinerja, hasilnya metode ini dapat memperoleh hasil rekomendasi yang memuaskan berbeda dengan sistem rekomendasi tradisional. Misalnya, *Restricted Boltzmann Machines* (RBM) yang terdiri dari *hidden layer* dan *visible layer*[14]*.*

#### Deep Collaborative Filtering

Algoritma *deep collaborative filtering* adalah algoritma yang dibuat untuk mengatasi masalah *sparsity* matriks *rating* pada algoritma *collaborative filtering.* Penyelesaian permasalahan tersebut dilakukan dengan mempelajari *side information* yang diperoleh dari profil *user* / *item* seperti demografi *user*, genre *item*, dll. *Deep collaborative filtering* yang menjembatanimatriks *factorization* dan *deep feature learning* merupakan *hybrid collaborative filtering model*. *Deep collaborative filtering* menggabungkan *probabilistic matrix factorization* dan *marginalized denoising auto-encoders. Probabilistic matrix factorization* adalah pendekatan *collaborative filtering* yang diterapkan secara luas dengan kinerja yang sangat baik, dan *marginalized denoising auto-encoders* adalah metode yang andal dalam mengekstraksi fitur tingkat tinggi dari *input*. *Marginalized denoising auto-encoders* merupakan varian dari *auto-encoder* yang merupakan metode *unsupervised learning.* Kombinasi keduanya memanfaatkan fungsinya masing-masing untuk mempelajari model yang lebih kompleks[8]. Notasi yang digunakan pada *deep collaborative filtering* dapat dilihat pada tabel 2.1 dan ilustrasi penerapan *deep collaborative filtering* dapat dilihat pada gambar 2.1.

Tabel 2.1 Ringkasan Notasi

|  |  |
| --- | --- |
| Notasi | Deskripsi |
| m | Jumlah *user* |
| n | Jumlah *item* |
| d | Dimensi *latent factor* |
| p | Dimensi *user feature* |
| q | Dimensi *item feature* |
| *R* € | Matriks *Rating* |
| *U* € | *Latent factor user* |
| *V* € | *Latent factor item* |
| *X* € | *Side information of user* |
| *Y* € | *Side information of item* |
| *W1*  € | *Mapping Function* untuk X di *auto-encoder* |
| *P1* € | Matriks *Projection* untuk *U* |



Gambar 2.1 Ilustrasi Penerapan *Deep Collaborative Filtering*

Inti dari *machine learning* adalah model yang dilatih dengan data pelatihan, dan metode untuk menyesuaikan parameter model adalah berdasarkan *loss function*. Tujuan dari pelatihan ini adalah untuk meminimalkan *loss* akibat salah prediksi rata-rata model. *Loss function* diterapkan untuk menilai kualitas dan kapabilitas model pada pelatihan model *machine learning*[15]. *Deep collaborative filtering* memanfaatkan *loss function* dari *probabilistic matrix factorization* untuk menguraikan matriks *rating* R yaitu, . Berikut fungsi objektif yang digunakan pada *deep collaborative filtering*:

(1)

Di mana:

A adalah matriks indikator yang menunjukkan entri yang tidak kosong di matriks *rating* dan melambangkan produk *hadamard* atau *pointwise product*. adalah pemetaan rekonstruktif, adalah matriks *Projection*, adalah parameter *tradeoff*. Istilah pertama menunjukkan proses pembelajaran *marginalized denoising auto encoders.* Ini mengukur kesalahan rekonstruksi antara input fitur *user* dan fitur yang dipetakan dari *input* yang rusak yakni, . adalah *learned mapping* yang diharapkan untuk meminimalkan *loss data.* Istilah kedua berfungsi untuk menghubungkan *hidden layer feature* dan *latent factor* U[8].

Tahapan Rekomendasi Pada Deep Collaborative Filtering :

* + - 1. Mencari *latent factor user* dan *item*

*Latent factor* *model* telah menjadi teknik yang sukses digunakan untuk membangun sistem rekomendasi. Sebagian besar penelitian difokuskan untuk mempelajari l*atent factor* dalam situasi data *sparsity*[16]. Model *latent factor* mencoba menemukan perkiraan *rating* di matriks *user-item*, di mana *rating* digunakan untuk mengisi entri yang hilang[17]. Pada proses untuk mendapatkan *latent factor* akan terdapat *noisy data* yang tak terhingga jumlahnya sehingga perlu dilakukan *mapping function* untuk *side information user* dan *item*[8]. Dilakukan *mapping function* W1 untuk *side information* *user* dan W2 untuk *side information item*.

(2)

(3)

,

.

Pada dan Misalkan = Z, dan p = *corrupt\_ratio*, untuk mendapatkan nilai Z setiap nilai yang bukan di diagonal akan dikalikan dengan (1 - p)2 dan yang berada di diagonal akan dikalikan dengan (1 - p)[18].

(4)

Demikian pula dengan dan , setiap elemen pada akan dikalikan dengan (1 - p).

(5)

Lalu diperlukan suatu matriks *projection* untuk memetakan *latent factor user* dan *item* ke *feature space.* *Feature space* merupakan kumpulan fitur yang digunakan untuk mengkarakterisasi data yang akan dilatih[19]. Berikut rumus matriks *projection*:

(6)

(7)

Di mana untuk matriks *projection* *latent factor user* dan untuk matriks *projection* *latent factor item*.

Untuk menyelesaikan permasalahan *latent factor* U dan V, digunakan algoritma *stochastic gradient descent (SGD).* SGD adalah metode yang befungsi untuk mengoptimalkan fungsi *objective* *machine-learning*[20]. Berikut rumus untuk mencari SGD:

(8)

(9)

Di mana η adalah *learning rate*, dan turunan dari rumus diatas didefinisikan sebagai berikut:

(10)

(11)

**Algoritma DCF untuk menemukan *latent factor* *user* dan *item***

**Input:** Matriks *rating* *R*, *user feature* *X, item feature* *Y,* *Parameters*

**Output :** *Latent factor* *U,V*

1: *Initialize* *U,V,*and

2: ***while*** *validation error decreases*, **do**

3: *Update* *using* (10);

4: *Update* *using* (11);

5: *Update* *using* (14);

6: *Update* *using* (15);

7: ***for*** *each* *observed* , ***do***

8: *Update* *using* (16);

9: *Update* *using* (17);

10: ***end for***

11: ***end while***

* + - 1. Menghasilkan rekomendasi *set*

Setelah *latent factor user* dan *item* ditemukan, dilakukan perkalian matriks untuk kedua *latent factor* seperti pada gambar 2.2, sehingga didapatkan *user-item* matriks *rating* yang baru. Matriks *rating* yang baru dapat dilihat pada tabel 2.2.



Gambar 2.2 Perkalian Matriks *Latent Factor User* dan *Item*

Tabel 2.2 Matriks *Rating* Hasil Prediksi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Item1 | Item2 | Item3 | Item4 |
| User1 | 2,36704 | 3,55596 | 1,68560 | 2,30801 |
| User2 | 3,61134 | 4,85569 | 3,06585 | 3,81974 |
| User3 | 3,13898 | 3,82720 | 2,66200 | 3,31269 |
| User4 | 2,88840 | 3,23815 | 2,30140 | 3,13117 |

Matriks *rating* yang didapatkan akan dilakukan normalisasi kembali menggunakan metode *min-max normalization* sehingga nilai *rating* yang didapat berada di dalam skala 1-10. Berikut rumus metode *min-max normalization*[21].

(12)

Di mana:

adalah nilai hasil normalisasi

adalah nilai *rating* yang lama

adalah nilai minimal pada matriks *rating*

adalah nilai maksimal pada matriks *rating*

adalah nilai minimal yang baru

adalah nilai maksimal yang baru

Normalisasi matriks *rating* menggunakan *min-max normalization* dapat dilihat pada tabel 2.3.

Tabel 2.3 Matriks *Rating* Prediksi Setelah Normalisasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Item1 | Item2 | Item3 | Item4 |
| User1 | 2,934633 | 6,31002 | 1 | 2,767044 |
| User2 | 6,467245 | 10 | 4,91858 | 7,058901 |
| User3 | 5,126198 | 7,08008 | 3,772035 | 5,619367 |
| User4 | 4,414793 | 5,407746 | 2,748278 | 5,104025 |

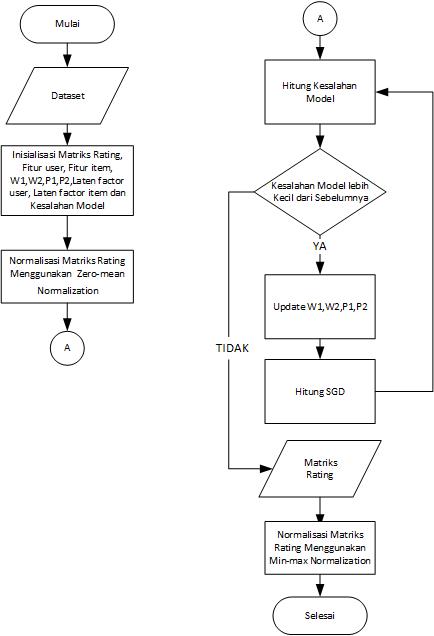
Jadi, setiap *user* akan mendapat rekomendasi *item* berdasarkan prediksi *rating user* terhadap *item* di mulai dari *item* dengan *rating* tertinggi sampai yang terendah.

### Analisis Proses

Analisis proses digunakan untuk menjelaskan proses kerja pada perangkat lunak untuk menyelesaikan permasalahan yang ada, yaitu rekomendasi *item* dengan *deep collaborative filtering*.Proses ini dimulai dari membaca *dataset* dari *database* kemudian proses rekomendasi *item* digital dengan *deep collaborative filtering*.

#### Deep Collaborative Filtering

Analisis proses cara kerja *deep collaborative filtering* dalam fitur rekomendasi *item* digital untuk memberikan rekomendasi *item* digital yang sesuai berdasarkan *rating* yang sudah diberikan oleh *user* dan juga berdasarkan profil *user* dan *item*. Penjelasan lebih lanjut mengenai alur proses dalam aplikasi sistem rekomendasi ini dijelaskan dengan *flowchart* berikut:



Gambar 3.1 *Flowchart* dari *Deep Collaborative Filtering*

## Pengujian dan Pembahasan

### Pengujian

Pada bagian ini akan ditampilkan hasil pengujian terhadap program dengan menggunakan algoritma *deep collaborative filtering* yang terdiri atas 2 (dua) bagian, yaitu pengujian perangkat lunak dengan *black box testing,* dan pengujian keakuratan hasil rekomendasi.

#### Pengujian Perangkat Lunak Dengan Black Box Testing

Berdasarkan hasil pengujian dengan kasus uji perangkat lunak dengan *black box testing* di atas dapat ditarik kesimpulan bahwa perangkat lunak secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan.

#### Pengujian Keakuratan Hasil Rekomendasi

Rekomendasi yang diberikan oleh sistem dipengaruhi oleh nilai *rating* yang diberikan oleh *user* terhadap sebuah *item* dan juga berdasarkan profil *user* dan *item*. Untuk dapat melihat sejauh mana keakuratan rekomendasi yang diberikan, maka perlu dilakukan pengujian terhadap nilai prediksi yang dihasilkan. Pengujian menggunakan *Mean Absolute Error* (MAE) dan *Root* *Mean* *Square* *Error* (RMSE) untuk menghitung rata-rata *error* dari nilai prediksi yang dihasilkan.

Pengujian dilakukan dengan menghitung hasil akhir dari jumlah *user* dan *item* seperti dibawah:

Jumlah *user* = 30

Jumlah *item* = 100

Jumlah data *rating* kosong = 2847

Total data = 3000

Total *absolute* MAE = 1904,07043457

Nilai *error* MAE = 1904,07043457 / 3000

= 0,6347

Total *absolute* kuadrat RMSE = 1693,77819824

Nilai *error* RMSE =

= 0,7514

Hasil pengujian untuk 30 *user,* 100 *item* dan 2847 data *rating* kosong menghasilkan nilai *error* MAE = 0,6347 dan nilai error RMSE = 0,7514.

### Pembahasan

Pengujian dengan kasus uji perangkat lunak dengan b*lack box testing* di atas dapat ditarik kesimpulan bahwa perangkat lunak secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan.

Dari hasil pengujian menggunakan MAE dan RMSE dari hasil prediksi yang dihasilkan menunjukkan bahwa nilai error yang didapatkan melalui penggunaan algoritma *deep collaborative filtering* pada 30 *user,* 100 *item* dan 2847 data *rating* kosong menghasilkan nilai rata-rata yang relatif rendah yaitu nilai *error* MAE = 0,6599 dan nilai error RMSE = 0,7753, dimana jika nilai mendekati nilai 1 maka semakin besar nilai error dan akurasi nya semakin rendah, sehingga dapat disimpulkan bahwa algoritma *deep collaborative filtering* memiliki keakuratan yang baik.

## Kesimpulan

Berdasarkan Pengujian yang dilakukan terhadap pengembangan Aplikasi Rekomendasi Item Pada Perpustakaan Digital Menggunakan *Deep Collaborative Filtering* Berbasis *Mobile* dan *Web* diperoleh beberapa kesimpulan yaitu sebagai berikut:

1. Berdasarkan hasil pengujian perangkat lunak dengan *black box testing* dapat ditarik kesimpulan bahwa perangkat lunak secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan.
2. Penggunaan metode *deep collaborative filtering* dalam menghasilkan rekomendasi, memiliki keakuratan yang baik dengan pengujian data terakhir terhadap 30 *user* dan 100 *item* dengan 2847 jumlah data *rating* kosongmenghasilkan nilai rata-rata *error* dengan pengujian MAE sebesar 0,6347 dan RMSE sebesar 0,7514. Berdasarkan nilai rata-rata *error* tersebut dapat disimpulkan bahwa *deep collaborative filtering* dapat mengurangi *sparsity* matriks *rating* pada perpustakaan digital.

## Saran

Saran dari penulis untuk pengembangan Aplikasi Rekomendasi E-book Pada Perpustakaan Digital Menggunakan Deep Collaborative Filtering Berbasis Mobile dan Web adalah pada format perpustakaan digital dapat diperluas kebeberapa format digital lainnya seperti format epub dan lainya.

## DAFTAR PUSTAKA

[1] Zhang, M. & Yang, Z., 2019. GACOforRec: Session-Based Graph Convolutional Neural Networks Recommendation Model. *IEEE Access,* Volume 7, p. 114077.

[2] Isinkaye, F., Folajimi, Y. & Ojokoh, B., 2015. Recommendation Systems: Principles, Methods And Evaluation. *Egyptian Informatics Journal,* p. 4.

[3] Rahmawati, S., Nurjanah, D. & Rismala, R., 2018. Analisis dan Implementasi Pendekatan Hybrid untuk Sistem Rekomendasi dengan Metode Knowledge Based Recommender System dan Collaborative Filtering. *Indonesian Journal Of Computing,* 3(2), pp. 12-13.

[4] Saeed, M. & Mansoori, E. G., 2017. A Novel Fuzzy-Based Similarity Measure For Collaborative Filtering To Alleviate The Sparsity Problem. *School of Electrical and Computer Engineering,* Volume 14, p. 2.

[5] Wenige, L. & Ruhland, J., 2018. Retrieval By Recommendation: Using Lod Technologies To Improve Digital Library Search. *Int J Digit Libr,* p. 2.

[6] Liu, J. & Wu, C., 2017. Deep Learning Based Recommendation: A Survey. p. 452.

[7] Ikasari, D., Suhendra, A. & Farida, N., 2018. Metode Deep Learning Pada Sistem Rekomendasi : Review Paper. Volume 2, pp. 46-47.

[8] Li, S., Kawale, J. & Fu, Y., 2015. Deep Collaborative Filtering via Marginalized DenoisingAuto-encoder. pp. 812, 814.

[9] Aithal, D. P. S., 2016. Smart Library Models For Future Generations. *International Journal Of Engineering Research And Modern Education (IJERME),* 1(1), p. 695.

[10] Khiste, G. P., Deshmukh, D. R. K. & Awate, A. P., 2018. Literature Audit of 'Digital Library': an Overview. *Vidyawarta,* p. 403.

[11] Nahak, B. & Padhi, S., 2019. The Role Of Smart Library And Smart Librarian For E- Library Services. p. 89.

[12] Elahi, M., Ricci, F. & Rubens, N., 2016. A Survey Of Active Learning In Collaborative Filtering Recommender Systems. *Computer Science Review,* Volume 20.

[13] Kwanggi, K., 2016. Book Review: Deep Learning. *Healthcare Informatics Research,* Volume 22, p. 1.

[14] Zhou, W. et al., 2018. Deep Learning Modeling for Top-N Recommendation With Interests Exploring. Volume 6, p. 51442.

[15] Chen, S., Liu, X. & Li, B., 2018. A Cost-Sensitive Loss Function for Machine Learning. p. 1.

[16] Rao, V., V, R. K. & Padmanabhan, V., 2017. Divide and Transfer: Understanding Latent Factors for Recommendation Tasks. p. 1.

[17] Sammut, C. & Webb, G. I., 2011. Latent Factor Models and Matrix Factorizations. p. 1.

[18] Chen, M., Xu, Z. (., Weinberger, K. Q. & Sha, F., 2012. Marginalized Denoising Autoencoders for Domain Adaptation. p. 4.

[19] Mohamed, A. E., 2017. Comparative Study of Four Supervised Machine Learning Techniques for Classification. Volume 7, p. 3.

[20] Li, X. & Orabona, F., 2019. On the Convergence of Stochastic Gradient Descent with Adaptive Stepsizes. p. 1.

[21] Gajera, V., Shubham, Gupta, R. & Jana, P. K., 2016. An Effective Multi-Objective Task Scheduling Algorithm using Min-Max Normalization in Cloud Computing. p. 813.