TESIS

MESIN REKOMENDASI FILM MENGGUNAKAN METODE KEMIRIPAN GENRE BERBASIS COLLABORATIVE FILTERING

INDAH SURVYANA WAHYUDI
NRP. 2215206701

DOSEN PEMBIMBING
Mochamad Hariadi, ST., M.Sc., Ph.D
Dr.Ir. Achmad Affandi, DEA

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA - CIO
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

TESIS

Mesin Rekomendasi Film Menggunakan Metode Kemiripan Genre Berbasis *Collaborative Filtering*

INDAH SURVYANA WAHYUDI
NRP. 2215206701

DOSEN PEMBIMBING
Mochamad Hariadi, ST., M.Sc., Ph.D
Dr.Ir. Achmad Affandi DEA

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA – CIO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperolch gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

Oleh:

Indah Survyana Wahyudi
NRP. 2215206701

Tanggal Ujian : 6 Juni 2017
Periode Wisuda : September 2017

Disetujui oleh:

1. Mochamad Hariadi, ST., M.Sc., Ph.D (Pembimbing I)
   NIP: 196912001997031002
2. Dr.Ir. Achmad Affandi, DEA. (Pembimbing II)
   NIP: 196510141990021001
3. Dr. Ir. Endroyono, DEA (Penguji)
   NIP: 196504041991021001
4. Dr. Surya Sumpeno, ST., M.Sc. (Penguji)
   NIP: 196906131997021003
5. Dr. Eko Mulyanto Yuniarno, ST., MT. (Penguji)
   NIP: 196806011995121009
6. Dr. Istas Pratomo, ST. MT. (Penguji)
   NIP: 197903252003121001

Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**MESIN REKOMENDASI FILM MENGGUNAKAN METODE KEMIRIPAN GENRE BERBASIS COLLABORATIVE FILTERING**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 4 Mei 2017

Indah Survyana Wahyudi
NRP. 2215206701

Halaman ini sengaja dikosongkan

Mesin Rekomendasi Film Menggunakan Metode Kemiripan Genre Berbasis Collaborative Filtering

Nama mahasiswa : Indah Survyana Wahyudi
NRP : 2215206701
Pembimbing :1.Mochamad Hariadi, ST., M.Sc., Ph.D
2.Dr.Ir. Achmad Affandi, DEA.

ABSTRAK

Era digital ditandai dengan *information overload* membutuhkan cara penemuan kembali informasi yang efektif. Sistem rekomendasi muncul sebagai solusi memberikan informasi atau item yang bersifat personal dengan mempelajari interaksi seorang pengguna dan pengguna lain dan items yang telah terekam sebelumnya tanpa memasukkan query seperti pada *search engine*. *Collaborative filtering* sebagai metode dalam memberikan rekomendasi personal. Pada paper ini peneliti menyodorkan sebuah model mesin rekomendasi untuk user baru dengan metode *Collaborative Filtering* dengan algoritma alternating least square-weight regularization (ALS-WR) yang kemudian di filter kembali berdasarkan kemiripian genrenya yang menggunakan algoritma *cosine similarity* dengan tujuan memberikan error terkecil dengan presisi yang tinggi. Untuk dataset peneliti menggunakan dataset dari movielens.org. Root Mean Squared Error (RMSE) pada saat training mendapatkan hasil prediksi pada dataset 100K adalah 0.96 (validasi) sementara 0.94 (test), pada dataset 1M nilai RMSE 0.86 (validasi) dan 0.96 (test), pada dataset 10M nilai RMSE 0.81 (validasi) sementara RMSE pada data test diperoleh 0.81 (test). Terlihat bahwa algoritma ALS-WR dapat mengatasi overfitting karena hasil dari validasi dan test pada saat training adalah sama. Terlihat juga semakin besar data, RMSE semakin kecil, dengan demikian ALS-WR dapat digunakan untuk data yang terus tumbuh dan bertambah. Hasil dari *cosine similarity* untuk mendekatkan hasil *collaborative filtering* dengan genrenya juga didapatkan nilai 1 untuk kemiripan 100% dan nilai itu akan berkurang berdasarkan tingkat kemiripan suatu item film yang dipilih user. Untuk uji penerimaan user didapatkan hanya 28% dari hasil rekomendasi pertama yang dapat diterima user, nilai ini meningkat menjadi 62% tingkat penerimaan user terhadap rekomendasi kedua. Hasil akhir ternyata 75% responden lebih menyukai rekomendasi kedua yaitu hasil dari filtering dua tahap dibandingkan hanya collaborative filtering saja.

Kata kunci: Mesin Rekomendasi; ALS-WR; *Cosine Similiarity*; *Collaborative Filtering*; *Big Data;*

Halaman ini sengaja dikosongkan

Recommender Engine Using Cosine Similarity Base On Alternating Least Square -Weight Regularization

By : Indah Survyana Wahyudi
Student Identity Number : 2215206701
Supervisor(s) : 1. Mochamad Hariadi, ST., M.Sc., Ph.D
2. Dr.Ir. Achmad Affandi, DEA.

ABSTRACT

The digital era marked with information overload requires an effective way of rediscovery of information. The recommendation system emerges as a solution to discovery personal information by studying the interaction of a user and items that have been previously recorded without enter a query like the search engine. Collaborative filtering with alternating least square-weight regularization algorithm (ALS-WR) as a method of providing personal recommendations. In this paper the researcher presented a recommendation engine model for the new user with Collaborative Filtering method which was filtered back based on its genre resemblance using cosine similarity algorithm with the aim of giving the smallest error with high precision. For dataset researchers use the dataset from movielens.org. Root Mean Squared Error (RMSE) during training gets predicted results on 100K datasets is 0.96 (validation) and 0.94 (test), on 1M dataset RMSE 0.86 (validation) and 0.96 (test), on 10M dataset RMSE 0.81 (validation)and .81 (test). It can be seen that ALS-WR algorithm not overfit because the result of validation and test during training is the same. Also visible the larger the data, the smaller RMSE, thus ALS-WR can be used for data that continues to grow and grow. The result of cosine similarity to get closer to the collaborative filtering with the genre is also obtained value 1 for 100% resemblance and the value will decrease based on the similarity level of a selected item of the user. For user acceptance test only 28% of the first user acceptable recommendation, this value increased to 62% acceptance level of the user against the second recommendation. The final result turned out that 75% of respondents prefer the second recommendation is the result of two-stage filtering than just collaborative filtering alone.

Key words: Recommender Engine; ALS-WR; Cosine Similiarity; Collaborative Filtering; Big Data;

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillahirobbil'alamin, puji syukur atas segala limpahan nikmat dan karunia Allah SWT, Tuhan yang Maha Kuasa. Hanya dengan petunjuk, rahmat dan ridho-Nya, sehingga penulis dapat menyelesaikan tesis dengan judul "**MESIN REKOMENDASI FILM MENGGUNAKAN METODE KEMIRIPAN GENRE BERBASIS COLLABORATIVE FILTERING".**

Ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya saya sampaikan kepada yang terhormat Mochamad Hariadi, ST., M.Sc., Ph.D selaku pembimbing pertama dan Dr.Ir. Achmad Affandi, DEA selaku pembimbing kedua, yang dengan penuh perhatian, dan kesabaran selalu meluangkan waktu, memberikan pengarahan dan motivasi serta semangat dalam penulisan tesis ini.

Penulis dapat menyelesaikan tesis ini, juga tidak terlepas dari bantuan dan kerjasama dari berbagai pihak, maka perkenankan saya dengan sepenuh hati menyampaikan terima kasih yang tak terhingga kepada:

1. Prof. Ir. Joni Hermana, M.Sc.Es, Ph.D., selaku Rektor Institut Teknologi Sepuluh Nopember Surabaya, yang telah memberikan kesempatan dan fasilitas kepada saya untuk mengikuti dan menyelesaikan pendidikan pada Program Magister, Jurusan Teknik Elektro, Bidang Keahlian Telematika / *Chief Information Officer* (CIO), Institut Teknologi Sepuluh Nopember Surabaya.
2. Kementerian Komunikasi dan Informatika Republik Indonesia yang telah memberikan kesempatan mendapatkan beasiswa Program Magister Jurusan Teknik Elektro, Bidang Keahlian Telematika / *Chief Information Officer* (CIO) pada Institut Teknologi Sepuluh Nopember Surabaya.
3. Dr. Adhi Dharma Wibawa, S.T., M.T., selaku Koordinator Bidang Keahlian Telematika / *Chief Information Officer* (CIO) sekaligus Dosen Pembimbing Akademik Program Magister (S2) Jurusan Teknik Elektro, Bidang Keahlian Telematika / *Chief Information Officer* (CIO) Angkatan Tahun 2015, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember atas arahan,

bimbingan dan motivasinya dalam menyelesaikan perkuliahan maupun penulisan tesis ini.

4. Seluruh Pengajar dan staf Program Studi Magister (S2) Jurusan Teknik Elektro, Bidang Keahlian Telematika / *Chief Information Officer* (CIO), yang telah mentransfer ilmu pengetahuannya melalui kegiatan perkuliahan maupun praktikum serta membantu kelancaran pengurusan administrasi perkuliahan dan penyelesaian tesis ini.
5. Orangt tua Penulis Bapak Wahyudi Sapto dan Ibu Hafsah terimakasih atas segala do'a dan dukungannya sehingga penulis dapat menyelesaikan tesis ini tepat waktu.
6. Suamiku tersayang Wawan Dedi Marahendra dan putriku tersayang Quinza Azqiara Lafatunnisa yang dengan penuh kesabaran, mendukung dan mendoakan demi selesainya studi ini. Semoga kita semua selalu mendapat ridlo-Nya dalam keberkahan. Aamiin.
7. Mahasiswa Program Studi Magister (S2) Telematika / *Chief Information Officer* (CIO) Angkatan 2015 yang selalu kompak dan saling mendukung, saling mendoakan baik dalam perkuliahan maupun dalam penyelesaian penulisan tesis ini.

Surabaya, Mei 2017

Penulis

DAFTAR ISI

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Era Digital dimana paradigma dan kebutuhan masyarakat dunia mulai bergeser dari informasi yang dibatasi oleh lembaran kertas menjadi informasi dalam bentuk digital. Internet menghidupkan sumber informasi digital dengan kemudahan akses dari berbagai perangkat teknologi informasi dan komunikasi secara cerdas, praktis dan terintegrasi. Terjadilah ledakan data dan informasi digital dimana social media, proses bisnis, publikasi-publikasi digital berperan meningkatkan volume dan pertumbuhan, tercatat 2,5 Exabyte data diciptakan setiap harinya, Google memproses 3,5 miliar permintaan dan menyimpan data 10 Exabyte di akhir 2016. Facebook sendiri memiliki 2,5 miliar konten, 2,7 miliar 'suka' dan 300 juta foto - yang semuanya jika dijumlahkan menjadi lebih dari 500 Terabyte data per hari dan tercatat tahun 2012 facebook telah menyimpan 100 Petabytes dari foto dan video. Amazon menyimpan data 1 Exabyte dari history dari jutaan pembeli untuk memprediksi *shopping needs*. Netflix perusahaan rental film yang merubah fungsi layanannya menjadi streaming video tahun 2016 tercatat memiliki 86.7 juta pelanggan yang berada di seluruh dunia dengan master Netflix Catalog yang memakan sekitar 3.14 Petabyte ruang penyimpanan *cloud* [1].

Ledakan data dan informasi memunculkan permasalahan penemuan kembali informasi (*information retrieval*). Mesin pencari (*search engine*) memecahkan sebagian masalah itu, namun personalisasi informasi tidak diberikan karena mesin pencari terbatas pada kata/*query* yang dimasukkan pengguna, persoalan muncul ketika pengguna mengetahui produk yang dibutuhkan tapi tidak mengetahui nama produk tersebut atau kata yang mewakili produk tersebut, seperti berkeliling di salah satu mall besar atau perpustakkaan besar dimana setelah berputar seharian tidak menemukan barang yang dicari. Rekomendasi dari seorang ahli yang mengerti kebutuhan dan profil pengguna menjadi sangat dibutuhkan. Disinilah era digital membawa era *Big Data* meninggalkan popularitas mesin pencari menuju mesin rekomendasi [2]. Sistem rekomendasi (*recommender system*) adalah sistem untuk penyaringan, pemilahan item dan informasi yang mengambil preferensi dari perilaku

pengguna, profil pengguna atau pendapat dari komunitas pengguna untuk membantu individu dalam mengidentifikasi konten yang menarik dan berpotensi besar untuk dipilih, dibeli atau digunakan [3], sementara mesin rekomendasi adalah inti dari sistem rekomendasi yang fokus pada penggunaan algoritma dan perhitungan matematika untuk mempelajari profil user, profil item dan interaksi keduanya dalam memprediksi item yang akan direkomenasikan kepada user [4]. Perbedaan hasil utama dari pencarian dan rekomendasi yaitu pencarian adalah daftar dokumen yang berkaitan atau sama dengan kata/*term*/*query* yang pengguna masukkan pada mesin pencarian, sementara hasil dari rekomendasi adalah seperti "*discovery*" sesuatu yang tidak terduga yang user tidak meduganya tetapi ternyata itu ada [5]. Fakta mengatakan 75% pelanggan Netflix didapatkan dari rekomendasi [6] , 89% pendapatan google diperoleh dari *Personal Advertise* yang diperoleh dari mesin rekomendasi dan pendapatan amazon naik 29% berkat mesin rekomendasi [7] .

E-commerce dan penyedia layanan jasa berbasis internet berlomba-lomba memperbaharui dan meningkatkan kualitas mesin rekomendasinya agar dapat menghasilkan rekomendasi terbaik bagi pengguna dan dapat mempengaruhi pengguna untuk membeli dan mengambil produk tersebut. Netflix mempekerjakan 300 orang dan menghabiskan 150 juta dollar pertahun khusus untuk memelihara dan meningkatkan kualitas mesin rekomendasi [8]. Sejak internet mulai populer digunakan tahun 1991, mesin rekomendasi mulai dikembangkan hingga era *Big Data* saat ini, mesin rekomendasi menjadi teknologi utama dalam pemasaran [9].

Beberapa metode populer yang digunakan dalam membuat mesin rekomendasi yaitu *content based filtering*, *collaborative filtering* dan *hybrid*. *content based filtering* memanfaatkan interaksi antara kontent item dengan profil pengguna [10] , dimana yang termasuk konten item disini seperti *genre*, *author*, dll. *collaboratif filtering* (CF) bekerja dengan membangun *database* preferensi konsumen terhadap suatu item. Seperti promosi dari mulut ke mulut, *collaborative filtering* memberikan prediksi rating dan personal rekomendasi berdasarkan yang disukai pengguna lain yang mempunyai selera yang sama [11]. Sementara *hybrid* merupakan penggabungan *content based filtering* dan *collaborative filtering*. Diantara ketiga metode diatas, perusahaan-perusahaan *e-commerce* raksasa seperti

Amazon, netflix, pandora menggunakan metode *collaborative filtering* sebagai representasi dari *wisdom of crowd*.

Namun ada tiga permasalahan yang besar dari *collaboratif filtering*, masalah yang pertama adalah data *sparsity* dimana banyak sistem recommender komersial didasarkan pada dataset besar akibatnya, matriks user-item yang digunakan untuk *collaborative filtering* bisa sangat besar dan jarang, ini menjadi tantangan dalam hasil rekomendasi dan ini berakibat pada jumlah error dalam perhitungan matriks. Masalah yang kedua adalah skalabilitas yaitu besarnya data yang akan berpengaruh pada proses jalannya perhitungan untuk menghasilkan rekomendasi. Masalah ketiga adalah cold start problem dimana user baru belum pernah memberikan rating dan item baru yang belum mendapatkan rating mulai memasuki sistem [12].

Penelitian sebelumnya mengenai *collaborative filtering* menggunakan terbagi dalam 2 pendekatan yaitu dengan metode *memory-based collaborative filtering* dan *model-based collaborative filtering*. Contoh algoritma yang digunakan pada *memory-based collaborative filtering* adalah *pearson corellation* [13] algoritma *k-NearstNeighbors* (k-NN) [14] [15] akan tetapi error/RMSE masih diatas 1, dan terdapat permasalahan *overfiting*, scalability atau skalabilitas, sparsity atau sparsitas, dan cold start problem pada user baru. sedangkan *dengan model-based collaborative filtering* menggunakan *Singular Value Decomposition* (SVD) [16], Stochastic Gradient Descent (SGD) dan Alternating Least Square (ALS), overfiting dapat diselrsaikan dengan baik oleh ketiga algoritma ini namun perbandingan ketiganya ALS-WR memberikan performansi yang lebih baik diantara ketiga algoritma tersebut terutama untuk menyelesaikan *scalability* atau skalabilitas, *sparsity* atau sparsitas, dan *cold start problem* pada user baru [17-19].

Pada penelitian ini peneliti akan mencoba untuk membangun sebuah mesin rekomendasi untuk user baru dengan *model-based collaborative filtering* yang mengatasi kekurangan dari *memori-based filtering* yaitu pada masalah skalabilitas dan sparsitas, menggunakan mesin pembelajaran Big Data yaitu Mlib Apache Spark. Mesin Pembelajaran M.lib Apache Spark diketahui sebagai tools big data algoritma yang digunakan pada *collaborative filtering* adalah algoritma *Alternating Least Square- Weight Regularization* (ALS_WR) dan *cosine similiarity* untuk menghitung kemiripan genre atau biasa disebut genre *similiarity*. Penggunaan dua metode ini

tidak di mix seperti pada metode *hybrid* tetapi dibuat 2 tahap yang bertujuan agar pengguna tidak hanya mendapatkan rekomendasi film yang disukai oleh user lain, tetapi juga mendapatkan rekomendasi sesuai dengan gendre film yang disukai.

1.2 Rumusan Masalah

Data yang terus tumbuh memerlukan suatu mesin rekomendasi sebagai alternatif untuk menemukan kembali item yang dibutuhkan. Penemuan kembali item yang dibutuhkan dapat melalui *preference* user terhadap suatu item dalam bentuk penilaian/rating. Akan tetapi tidak semua item memperoleh rating dari user, terdapat matriks user dan item yang sangat jarang (*sparse*). Dengan *collaborative filtering* yang menggunakan ALS-WR dapat memprediksi rating untuk seluruh item untuk kemudian diolah kembali menjadi rekomendasi personal menurut penilaian seorang user. Namun, user terkadang juga menginginkan item terutama item film berdasarkan genrenya. Rumusan permasalahan adalah sebagai berikut :

1. Bagaimana implementasi metode *collaborative filtering* dengan algoritma ALS-WR untuk data berskala besar dalam menghasilkan rekomendasi?
2. Berapa tingkat error yang dihasilkan dalam memprediksi rekomendasi dan apakah sparsitas dapat ditanggulangi tanpa *overfitting* dalam memprediksi?
3. Bagaimana implementasi untuk menghasilkan rekomendasi dengan *collaborative filtering* dengan kemiripan genre?

1.3 Tujuan

Membangun mesin rekomendasi dari data berskala besar dengan merekomendasikan item berdasarkan prediksi rating dari keseluruhan pengguna terhadap suatu item namun tetap mempertahankan kemiripan genre pada item yang dipilih user.

1.4 Batasan Masalah

Dalam penelitian ini terdapat beberapa batasan masalah :

1. Hanya menggunakan teknik *item based collaborative filtering* karena pada teknik ini item yang pernah dirating pengguna akan menjadi landasan proses rekomendasi
2. Pengujian mesin dilakukan di *localhost* tidak pada *web service* berbayar

3. Menggunakan dataset dari *movielends.org* dengan jumlah rating 100K, 1M dan 10M
4. Dari keseluruhan fitur konten pada item, fitur yang diambil untuk rekomendasi tahap kedua ini adalah hanya fitur genre

1.5 Kontribusi

Menghasilkan mesin rekomendasi untuk data berskala besar dengan merekomendasikan item berdasarkan prediksi rating dari keseluruhan pengguna terhadap suatu item namun tetap mempertahankan kemiripan genre pada item yang dipilih user

1.6 Metodologi Penelitian

Bab I : Bab ini merupakan bagian yang akan menguraikan latar belakang, rumusan masalah, tujuan, batasan masalah, kontribusi, dann metodologi penelitian

Bab II : Bab ini berisi tentang landasan teori yang digunakan membangun mesin rekomendasi dan menjawab permasalahan penelitian.

Bab III : Bab ini berisi tentang metodologi dan metode yang digunakan untuk membangun dan menguji mesin rekomendasi yang dihasilkan penelitian ini.

Bab IV : Bab ini berisi tentang hasil yang didapatkan dari model mesin rekomendasi yang dibangun ini pada penelitian serta evaluasi dan analisis terhadap hasil dan evaluasi yang didapatkan.

Bab V : Bab ini berisi kesimpulan dan saran mengenai hasil evaluasi dan anlisis serta penelitian selanjutnya yang dapat memperaiki hasil penelitian ini

Halaman ini sengaja dikosongkan

BAB 2
KAJIAN PUSTAKA

2.1 *Big Data*

2,5 exabyte data atau 2,5 triliun byte data Setiap hari, diciptakan. Data ini berasal dari berbagai macam sumber, sensor digunakan untuk mengumpulkan informasi iklim, posting ke situs media sosial, gambar digital dan video, catatan transaksi pembelian,dll. Data ini adalah *big data*. Menurut [33] big data merupakan istilah yang berlaku untuk informasi yang tidak dapat diproses atau dianalisis menggunakan alat tradisional. Menurut [34] , big data adalah data yang melebihi proses kapasitas dari kovensi sistem database yang ada. Data terlalu besar dan terlalu cepat atau tidak sesuai dengan struktur arsitektur database yang ada. Untuk mendapatkan nilai dari data, maka harus memilih jalan altenatif untuk memprosesnya.

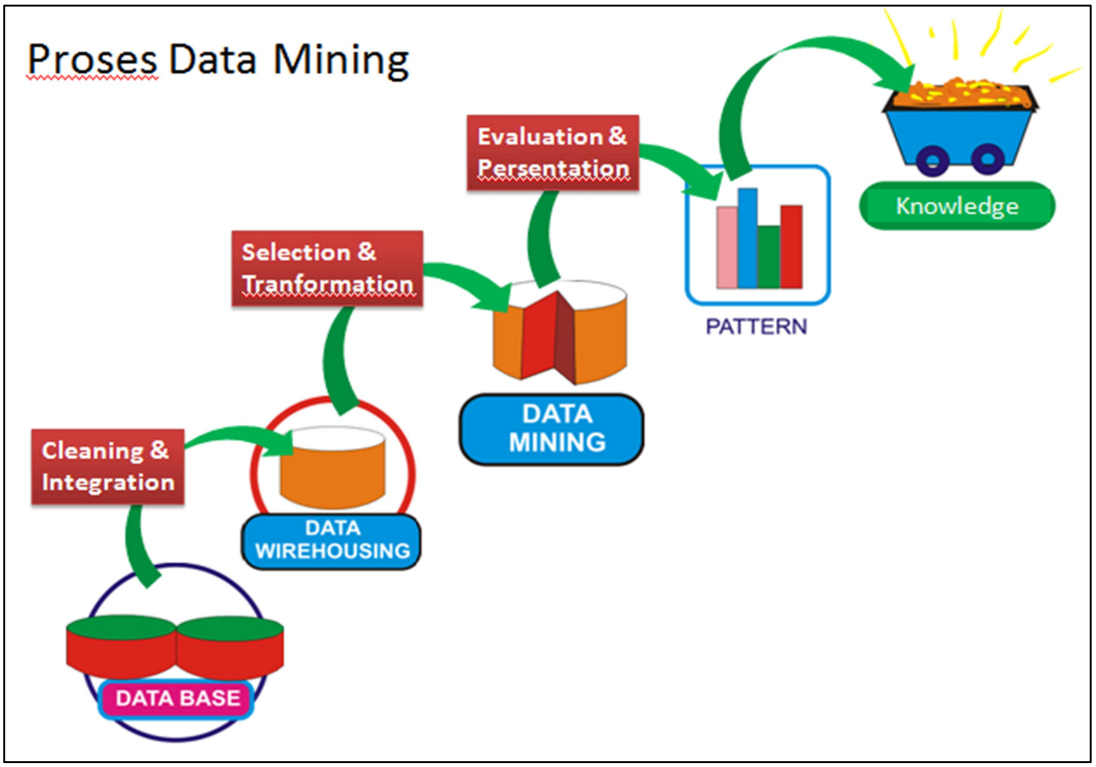
Big data dapat juga didefinisikan sebagai sebuah masalah domain dimana teknologi tradisional seperti relasional database tidak mampu lagi untuk melayani. Big data lebih dari hanya masalah ukuran, *Big data* adalah kesempatan untuk menemukan wawasan dalam jenis baru dan muncul data dan konten, untuk membuat proses bisnis lebih cepat, dan menjawab pertanyaan yang sebelumnya dianggap di luar jangkauan. Namun 3 dimensi big data yang merupakan inti dari masalah yang sampai saat ini masih merupakan kajian untuk sousinya yaitu *Volume, Variety dan Velocity*. *Volume* menandakan bahwa ukuran dan kapasitas data akan terus tumbuh dan bertambah seiring dengan pertambahan waktu. *Variety* adalah keragaman yang berarti bahwa kategori yang dimiliki big data dan juga fakta yang sangat penting untuk dianalisis. *Velocity* adalah kecepatan, yang mengacu pada kecepatan data atau seberapa cepat data dihasilkan dan diproses untuk memenuhi tuntutan dan tantangan dijalur pertumbuhan dan perkembangan

Keterbatasan *big data* karena set data yang besar. *Data set* tumbuh dengan cepat dan saling terkait. Kemampuan analisis *big data* diyakini mampu membantu untuk mengatasi ragam persoalan. Bagi pelaku bisnis, insights yang diperoleh akan bermanfaat untuk mengambil keputusan yang cepat dan akurat: kapan melempar

produk ke pasar, promosi seperti apa yang tepat untuk pasar yang ditargetkan, bagaimana menghadirkan rekomendasi yang sesuai dengan karakter dan minat pelanggan, hingga bagaimana mengatasi keluhan konsumen sebelum menularkan kabar buruk dengan cepat bagaikan virus.

2.2 Data Mining untuk Analisis Big Data

Kemampun big data analytics merupakan keunggulan yang dicari di tengah kompetisi bisnis yang ketat. Dalam melakukan data analytic memerlukan teknik data mining. Data mining adalah kegiatan mengekstraksi, menemukan hubungan, pola, dan kecendrungan dari data yang berukuran/berjumlah besar. Data mining juga dikenal dengan nama *Knowledge Discovery in Databases* (KDD). Mesin pembelajaran (learning machine) menjadi salah satu alat dalam data mining untuk menemukan pola, menjalankan statistika, matematika dan kecerdasan buatan dalam mengekstraksi dan mengidentifikasi pengetahuan yang bermanfaat pada suatu database yang besar. langkah-langkah untuk melakukan data mining terdapat pada Gambar 2.1



Gambar 2.1 Langkah-langkah Proses *Data Mining*

1. *Data cleaning* (untuk menghilangkan noise data yang tidak konsisten). Data integration (di mana sumber data yang terpecah dapat disatukan)
2. *Data selection* (di mana data yang relevan dengan tugas analisis dikembalikan ke dalam database)
3. *Data transformation* (di mana data berubah atau bersatu menjadi bentuk yang tepat untuk menambang dengan ringkasan performa atau operasi agresi)
4. *Data mining* (proses esensial di mana metode yang intelejen digunakan untuk mengekstrak pola data)
5. *Pattern evolution* (untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik)
6. *Knowledge presentation* (di mana gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambang kpada user).

Secara umum tugas data mining dapat diklasifikasikan ke dalam dua kategori: deskriptif dan prediktif. Tugas menambang secara deskriptif adalah untuk mengklasifikasikan sifat umum suatu data di dalam database. Tugas data mining secara prediktif adalah untuk mengambil kesimpulan terhadap data terakhir untuk membuat prediksi.

1. Konsep/ *Description*

Data dapat diasosiasikan dengan pembagian class atau konsep. Untuk contohnya, ditoko All Electronics, pembagian class untuk barang yang akan dijual termasuk komputer dan printer, dan konsep untuk konsumen adalah big Spenders dan budget Spender. Hal tersebut sangat berguna untuk menggambarkan pembagian class secara individual dan konsep secara ringkas, laporan ringkas, dan juga pengaturan harga. Deskripsi suatu class atau konsep seperti itu disebut class/concept descripition.

2. *Association rules*

Association analysis adalah penemuan *association rules* yang menunjukkan nilai kondisi suatu *attribute* yang terjadi bersama-sama secara terus-menerus dalam memmberikan set data. *Association analysis* secara luas dipakai untuk market basket atau analisa data transaksi.

3. Klasifikasi dan Predikasi

Klasifikasi dan prediksi mungkin perlu diproses oleh analisis relevan, yang berusaha untuk mengidentifikasi atribut-atribut yang tidak ditambahkan pada proses klasifikasi dan prediksi. Atribut-atribut ini kemudian dapat di keluarkan.

4. *Cluster Analysis*

Tidak seperti klasifikasi dan prediksi, yang menganalisis objek data dengan kelas yang terlabeli, clustering menganalisis objek data tanpa mencari keterangan pada label kelas yang diketahui. Pada umumnya, label kelas tidak ditampilkan di dalam latihan data simply, karena mereka tidak tahu bagaimana memulainya. Clustering dapat digunakan untuk menghasilkan label-label.

5. *Outlier Analysis*

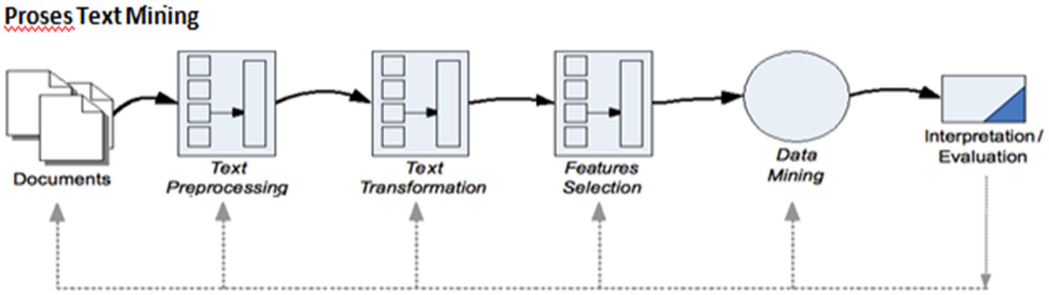
Outlier dapat dideteksi menggunakan test yang bersifat statistik yang mengambil sebuah distribusi atau probabilitas model untuk data, atau menggunakan langkah-langkah jarak jauh di mana objek yang penting jauh dari cluster lainnya dianggap outlier. Sebuah database mungkin mengandung objek data yang tidak mengikuti tingkah laku yang umum atau model dari data. data ini disebut outlier.

6. *Evolution Analysis*

Data analisa evolusi menggambarkan ketetapan model atau kecenderungan objek yang memiliki kebiasaan berubah setiap waktu. Meskipun ini mungkin termasuk karakteristik, diskriminasi, asosiasi, klasifikasi, atau clustering data berdasarkan waktu, kelebihan yang jelas seperti analisa termasuk analisa data time-series, urutan atau pencocockkan pola secara berkala, dan kesamaan berdasarkan analisa data

Text mining, yang juga disebut sebagai *Teks Data Mining* (TDM) atau *Knowledge Discovery in Text* (KDT), secara umum mengacu pada proses ekstraksi informasi dari dokumen-dokumen teks tak terstruktur (*unstructured)*. Tujuan utama *text mining* adalah mendukung proses *knowledge discovery* pada koleksi dokumen yang besar. *Text mining* mencoba memecahkan masalah information *overload* dengan menggunakan teknik-teknik dari bidang ilmu yang terkait. *Text mining* dapat dipandang sebagai suatu perluasan dari data mining atau *knowledge-discovery in database* (KDD), yang mencoba untuk menemukan pola-pola menarik dari basis data berskala besar. Perbedaan mendasar antara text mining dan data mining terletak pada

sumber data yang digunakan. Pada data mining, pola-pola diekstrak dari basis data yang terstruktur, sedangkan di *text mining*, pola-pola diekstrak dari data tekstual (*natural language*). *Text mining* merupakan suatu proses yang melibatkan beberapa area teknologi. Namun secara umum proses-proses pada *text mining* mengadopsi proses data mining. Bahkan beberapa teknik dalam proses *text mining* juga menggunakan teknik-teknik *data mining.* Ada empat tahap proses pokok dalam *text mining* yang terlihat pada gambar 2.2, yaitu pemrosesan awal terhadap teks (*text preprocessing*), transformasi teks (*text transformation*), pemilihan fitur (*feature selection*), dan penemuan pola (*pattern discovery*).



Gambar 2.2 Langkah-langkah Proses Text Mining

1. *Text Preprocessing*

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) terhadap teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Operasi yang dapat dilakukan pada tahap ini meliputi *part-of-speech* (PoS) tagging, menghasilkan parse tree untuk tiap-tiap kalimat, dan pembersihan teks.

2. *Text Transformation*

Transformasi teks atau pembentukan atribut mengacu pada proses untuk mendapatkan representasi dokumen yang diharapkan. Pendekatan representasi dokumen yang lazim digunakan oleh model "*bag of words*" dan model ruang vector (*vector space model*). Transformasi teks sekaligus juga melakukan pengubahan kata-kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen. Tindakan ini diwujudkan dengan menerapkan stemming dan menghapus *stop words*.

3. *Feature Selection*

Pemilihan fitur (kata) merupakan tahap lanjut dari pengurangan dimensi pada proses transformasi teks. Walaupun tahap sebelumnya sudah melakukan penghapusan katakata yang tidak deskriptif (*stopwords*), namun tidak semua kata-kata di dalam dokumen memiliki arti penting. Oleh karena itu, untuk mengurangi dimensi, pemilihan hanya dilakukan terhadap kata-kata yang relevan yang benar-benar merepresentasikan isi dari suatu dokumen. Ide dasar dari pemilihan fitur adalah menghapus kata-kata yang kemunculannya di suatu dokumen terlalu sedikit atau terlalu banyak. Algoritma yang digunakan pada *text mining*, biasanya tidak hanya melakukan perhitungan pada dokumen saja, tetapi juga pada feature . Empat macam feature yang sering digunakan:

• *Character*, merupakan komponan individual, bisa huruf, angka, karakter spesial dan spasi, merupakan block pembangun pada level paling tinggi pembentuk semantik feature, seperti kata, term dan concept. Pada umumnya, representasi *character-based* ini jarang digunakan pada beberapa teknik pemrosesan teks.

• *Words*.

• *Terms* merupakan single word dan *multiword phrase* yang terpilih secara langsung dari corpus. Representasi *term-based* dari dokumen tersusun dari subset term dalam dokumen.

• *Concept*, merupakan feature yang di-generate dari sebuah dokumen secara manual, *rule-based*, atau metodologi lain.

• *Pattern Discovery*

Pattern discovery merupakan tahap penting untuk menemukan pola atau pengetahuan (knowledge) dari keseluruhan teks. Tindakan yang lazim dilakukan pada tahap ini adalah operasi text mining, dan biasanya menggunakan teknik-teknik data mining. Dalam penemuan pola ini, proses text mining dikombinasikan dengan proses-proses data mining. Masukan awal dari proses text mining adalah suatu data teks dan menghasilkan keluaran berupa pola sebagai hasil interpretasi atau evaluasi. Apabila hasil keluaran dari penemuan pola belum sesuai untuk aplikasi, dilanjutkan evaluasi dengan melakukan iterasi ke satu atau beberapa tahap sebelumnya. Sebaliknya, hasil interpretasi merupakan tahap akhir dari proses text mining dan akan disajikan kepengguna dalam bentuk visual

2.3 Sistem Rekomendasi dan Mesin Rekomendasi

Sistem rekomendasi adalah sistem yang dirancang dengan tujuan untuk membantu pengguna dengan cara memberikan rekomendasi kepada pengguna ketika pengguna dihadapkan dengan jumlah informasi yang besar. Rekomendasi yang diberikan diharapkan dapat membantu pengguna dalam proses pengambilan keputusan, seperti barang apa yang akan dibeli, buku apa yang akan dibaca, atau musik apa yang akan didengar, dan lainnya [10]. Mesin rekomendasi adalah bagian dari sistem rekomendasi yang merupakan *core* atau otak dari sistem rekomendasi yang terfokus pada penggunaan algoritma dan perhitungan matematika untuk mempelajari profil pengguna, profil item dan interaksi keduanya dalam memprediksi item yang akan direkomendasikan kepada user [4] karena sebuah sistem rekomendasi dirancang untuk memprediksi sekumpulan item yang sesuai dengan preferensi user dimana nantinya item tersebut akan direkomendasikan pada user [20]. Mesin rekomendasi film adalah bagian dari sistem rekomendasi dari sebuah webpage

Profil pengguna dapat berisi tentang detail informasi pengguna seperti nama, umur, alamat, dsb. Profil item terdiri dari detail informasi tetang item beserta fiturnya seperti judul film, artis, sutradara, genre,dsb. Interaksi antara pengguna dan item adalah preferensi pengguna terhadap suatu item contohnya nilai suatu item yang diberikan pengguna, ketertarikan pengguna pada suatu item dan juga *history* interaksi antara pengguna dengan item [21].

Mesin rekomendasi pribadi (*personalized recommender engine*) harus mengenal terlebih dahulu setiap pengguna yang ada dengan membangun dan memelihara user model atau user preference yang berisi ketertarikan pengguna dengan item tertentu dengan metode dan algoritma yang telah ditetapkan [22]. Sebagai contoh, sistem rekomendasi di website Amazon menyimpan dan mempelajari setiap transaksi pembelian pelanggan, komentar pelanggan, dan review / rating yang diberikan oleh pelanggan terhadap suatu produk.

Terdapat dua buah dalam pengambilan data, yaitu pendekatan *implicit* dan *explicit*. Pendekatan *implicit*, artinya, sistem menyimpan dan mempelajari perilaku pengguna terhadap item, contohnya item apa yang pernah dibeli pengguna, berapa kali penggula melihat barang tersebut, dsb. Sementara pendekatan *explicit*, yaitu

dengan menanyakan kepada pengguna secara langsung deskripsi item yang bagaimana yang ia sukai/minati contoh keluarannya berupa rating atau kuesioner [23]. Mesin rekomendasi juga dapat mengambil preferensi dari *feedback* yang diberikan, bisa dijadikan satu-satunya preferensi atau preferensi kedua yang menjadi input atau aturan agar algoritma dapat bekerja atau dikontrol dengan baik. Contohnya dalam penelitian ini peneliti menggunakan *eksplisit preference dan implisit feedback*. *Explicit preference* berupa nilai yang diberikan pengguna terhadap suatu film dalam bentuk rating, dan *implisit feedback* adalah nilai yang diberikan pengguna dijadikan sebuah aturan atau regularisasi atau pinalti yang menandakan bahwa hanya item-item yang mendapat nilai pengguna yang di masukkan dalam perhitungan rekomendasi untuk hasil yang mengarah pada item yang terbaik (*Top N*) [24].

Berbagai metode telah ditemukan untuk menyediakan rekomendasi yang handal. Berdasarkan metode rekomendasi yang sering digunakan, sistem rekomendasi dibagi dalam tiga klasifikasi yaitu: *collaborative filtering recommendation, content-based recommendation ,* dan *hybrid.*

2.4 Mesin Rekomendasi Film

Mesin rekomendasi film adalah bagian dari sistem rekomendasi dari sebuah website gudang film atau website ecommerce yang bergerak dibidang usaha penyewaan film dmn sistemnya saat ini telah berubah dari peminjaman ke ijin akses untuk meelihat film secara realtime atau ijin pengunduhan fillm secara temporary. Mesin rekomendasi film juga menggunakan metode dan algoritma tertentu yang digunakan sebagai komputasi untuk menghasilkan rekomendasi yang bersifat personal. Berikut ini beberapa website gudang film dan website penyewaan film yang terkenal

1. **Netflix**

Netflix adalah pelopor layanan sewa film online. Didirikan sejak 1997, Netflix mengakomodasi arsip film paling lengkap dengan wilayah pengoperasian terbanyak. Namun, menyesuaikan dengan perkembangan teknologi, Netflix merubah layanan persewaan film menjadi layanan streaming yang memungkinkan pengguna menonton tayangan kesukaan di mana pun, kapan pun, dan hampir lewat medium apa pun (smartphone, smartTV, tablet, PC, dan laptop). Mirip langganan televisi berbayar (*cable tv*), Netflix bersih dari iklan, penonton tak perlu menunggu jadwal

penayangan serial televisi, dan bisa menentukan sendiri konten yang ingin dinikmati. Untuk pembuatan rekomendasi, netflix meminta peilaian pengguna berupa rating 1-5 sebagai feedback atas film yang pernah ditonton pengguna tersebut. Kemudian hasil dari rating tersebut dijadikan masukan untuk pembuatan personal rekomendasi. Khusus untuk memperbaharaui mesin rekomendasinya netflik mempekerjakan lebih dari 300 orang karena rekomendasi membawa peningkatan keuntungan Netflix. Kontribusi Netflix dalam bidang rekomendasi adalah netflix mengadakan sayembara untuk meningkatkan mesin rekomendasinya, namun hasil dari sayembara tersebut dipublikasikan secara umum. Metode dan algoritma yang paling terkenal adalah matriks faktorisasi collaboration filtering dalam memperingkat film berdasarkan rating dan prediksi rating yang dihasilkan algoritma tersebut. Sampai saat ini algoritma tersebut banyak di gunakan dan dikembangkan pada perusahaan ecommerce raksasa di dunia seperti Amazon dan yahoo.

2. **Rotten Tomatoes**

Rotten Tomatoes adalah situs web yang menyediakan informasi tentang film dari seluruh dunia, termasuk orang-orang yang terlibat di dalamnya mulai dari aktor/aktris, sutradara, penulis sampai penata rias dan musikus. Rotten Tomatoes juga dikenal sebagai review agregator (situs pengumpul review-review film dari situs lainnya) yang juga menyajikan berbagai ulasan terpercaya dari para kritikus film profesional dan penonton film (audience). Para kritikus film ini akan memberikan nilai terhadap sebuah film dalam Tomatometer. Sedangkan untuk para penonton biasa juga bisa menilai sebuah film melalui Audience Score. Ada banyak variabilitas dalam kualitas rekomendasi. Pada akhirnya penilian dan pemeringkatan film dari Rotten Tomatoes menjadi referensi bagi pengguna dalam memilih film terbaik yang ingin ditontonnya.

3. **Movielends**

Movielendsadalah situs rekomendasi film personal yang dususn berdasarkan rating yang diberikan pengguna terhadap sebuah film. MovieLens dijalankan oleh GroupLens, sebuah laboratorium penelitian di University of Minnesota. Dengan menggunakan MovieLens, pengguna turut membantu GroupLens mengembangkan alat dan antarmuka eksperimental baru untuk eksplorasi data dan rekomendasi. MovieLens tidak komersial, dan bebas dari iklanat. Movielends menerbitkan datasets

nya secara Cuma-Cuma untuk pengembangan dan penelitian bildang sistem rekomendasi. Penelitian terpublikasi yang terdeksi menggunakan dataset dari Movielends diindex dan dijadikan referensi bagi grup riset grouplends dan masyarakat yang tertarik pada pengembangan dan penelitian sistem rekomendasi.

4. **Flixster**

Seperti Netflix Flixster merupakan salah satu situs penyewaan, penjualan dan layanan nonton straming film. Flixter juga merupakan jejaring sosial bagi penggemar film di dunia. Di jejaring sosial ini, pengguna dapat melakukan berbagai hal, diantaranya membuat profil mereka sendiri, mengundang teman, melihat peringkat film beserta aktor, ulasan film dan memungkinkan pengguna untuk bertemu dengan orang lain yang mempunyai selera yang sama dengan mereka. Dari situs ini, pengguna juga bisa bercakap-cakap dengan pengguna lain, mendapatkan jadwal pemutaran film yang akan tayang atau yang sedang tayang, melihat foto selebriti populer yang mereka sukai, membaca berita film terbaru yang akan dirilis dan yang sedang dirilis, dan melihat klip video dari film populer. Dengan begitu banyak fitur, Flixster tampaknya telah menciptakan sebuah situs film jejaring sosial yang cukup komprehensif, yang pada gilirannya menjelaskan pertumbuhan yang cepat dalam popularitas perfilman. Biasanya para penggila film menggunakan layanan jejaring sosial ini untuk melihat ulasan film, mulai dari penggemar film horor, romantis, komedi. Selain itu, Flixster juga menyediakan berbagai macam permainan yang menarik untuk diikuti, ada kuis, polling yang tentunya berkaitan langsung dengan dunia perfilman. Pada bulan Januari 2010, Flixster membeli situs Rotten Tomatoes, sistem rekomendasi pada flixter kemudian menggunakan pemeringkatan dari Rotten Tomatoes.

5. **IMDb**

Internet Movie Database (IMDb) adalah situs web yang menyediakan informasi mengenai film dari seluruh dunia, termasuk orang-orang yang terlibat di dalamnya mulai dari aktor/aktris, sutradara, penulis sampai penata rias dan musikus. Situs web ini sekarang dimiliki oleh Amazon.com. Di dalam IMDB juga ada komunitas yang dapat berkontribusi langsung untuk menuangkan review tentang film dan memberikan rating pada film tersebut. Tidak hanya dari kaum awam, para pakar-pun juga mempunyai wadah sendiri untuk memberi rating dan menuangkan review secara

profesional pada film tersebut. Pengguna yang telah terdaftar (registered users) bisa memberikan penilaiannya atas suatu film dengan menggeser kursornya di atas rentetan bintang-bintang dari 0 sampai 10, atau memilih dari dropdown menu untuk situs mobile IMDB. Rating yang pengguna input akan memengaruhi nilai sebuah film. Nilai yang diinputkan seorang pengguna tidak langsung merubah nilai rating sebuah film namun jika dipopulasikan dengan registered user lainnya yang jumlahnya mencapai jutaan itu, tentu akan ada perubahan secara signifikan. menggunakan sistem weighted average (pembobotan). Banyak parameter yang diikutsertakan dalam formula IMDb untuk menghasilkan nilai akhir user rating sebuah film. Alasan di balik penggunaan sistem ini adalah untuk menghilangkan unsur ballot stuffing atau vote stuffing, yakni pemberian vote berkali-kali oleh satu user dengan mengubah-ubah vote yang sudah diberikan (karena pada dasarnya ketika satu user telah memberi vote, maka vote-nya akan tercatat di account IMDB-nya walaupun itu masih bisa diubah-ubah di kemudian hari). Ini menjadi penting karena banyak kasus di mana registered users memberi vote tidak untuk menyampaikan opini atau penilaian mereka terhadap suatu film, tetapi hanya untuk merubah user rating yang sudah ada. Fitur “Top 250” ini merekomendasikan 250 film di IMDB yang memiliki user rating tertinggi dari jutaan film lainnya dalam database IMDB. Film yang bisa masuk “Top 250” hanyalah full-length theatrical movies. Film pendek, film televisi (FTV), TV-series, atau film dokumenter tidak diikutsertakan dalam “Top 250”. Rating untuk “Top 250” didasarkan pada vote dari para regular voters. Regular voters adalah bagian dari registered users, tetapi parameter untuk menentukan apakah satu registered user merupakan regular voter atau tidak itu dirahasiakan oleh IMDB IMDB memiliki formula yang disadur dari teori Bayesian statistics yang dinilai memiliki kredibilitas yang cukup baik. IMDb juga menyediakan dataset untuk pengembangan ilmu sistem rekomendasi. Dataset IMDb dapat diperoleh dari situs Amazone Web Service https://aws.amazon.com/s3/.

6. **Criticker**

Critiker adalah adalah situs komunitas film dan rekomendasi film. situs ini membandingkan peringkat seorang pengguna dengan pengguna lain "Taste Compatibility Index" untuk melihat seberapa dekat selera seorang pengguna dengan

pengguna lain ini. Setelah layanan menemukan kecocokan, seorang pengguna dapat melihat profil pengguna lain dan melihat film mana yang mereka suka.

7. **Clerkdogs**

Clerkdogs adalah situs rekomendasi film. ClerkDogs mengandalkan database sendiri dari 18.000 film dan acara TV. CrekDogs dapat merekomendasi film favorit pengguna berdasarkan genrenya, rekomendasi film berdasarkan kesamaan selera antar pengguna, membandingkan dua atau lebih film secara berdampingan, glosarium untuk deskripsi kata kunci dan supermatch sebagai personal rekomendasi. ClerkDogs saat ini telah dijual kepada Netflix untuk memperkaya database dan sistem rekomendasi Netflix.

8. **Nanocrowd**

Nanocrowdadalah aplikasi rekomendasi film untuk pengguna iphone. Nanocrowd memungkinkan pengguna untuk memilih salah satu film kemudian dari pemilihan tersebut muncul beberapa daftar rekomendasi film yang dapat link ke Netflix atau Amazon untuk pemutaran film secara streaming.Dalam menciptakan daftar rekomendasi, Nanocrowd menciptakan kategori tiap film dengan menganalisis kata-kata yang digunakan orang lain di situs ulasan lain untuk menggambarkan film. Teknologi semantik digunakan Nanocrowd untuk menciptakan rekomendasi. Nanocrowd memungkinkan pengguna menelusuri pencarian yang lebih halus dengan bantuan " three-word nanogenre".

9. **Taste Kid**

Teste Kid yang berubah namanya menjadi TesteDive adalah mesin rekomendasi yang praktis. Tidak hanya berfungsi sebagai rekomendasi film tetapi rekomendasi buku, jenis band, dll yang berkaitan dengan film yang pengguna sukai. Jika seorang pengguna benar-benar menyukai "The Godfather," Taste Kid memberikan daftar rekomendasi berupa jenis band, buku dan "barang lain" yang terkait dengan film itu. Taste Kid adalah mesin rekomendasi hiburan penuh.

10. **Jinni**

Jinni adalah mesin pencari dan mesin rekomendasi untuk film, acara TV dan film pendek. Layanan ini didukung oleh Entertainment Genome, sebuah pendekatan untuk mengindeks judul berdasarkan atribut seperti mood, tone, plot, dan struktur. Ketersediaannya adalah melalui API, dalam perizinan business-to-business, di mana

hal itu mempengaruhi peningkatan keuntungan dalam bisnis. Layanan Jinni menggunakan teknik semantik, yaitu pendekatan berbasis makna untuk menafsirkan pertanyaan dengan mengidentifikasi konsep dalam konten, bukan kata kunci. Teknologi Jinni melibatkan taksonomi yang dibuat oleh para profesional film, dengan judul baru diindeks melalui metode Pengolahan Bahasa Alami dan Mesin untuk menganalisis secara otomatis tinjauan dan metadata. Mesin penemuan semantik Jinni didukung oleh Entertainment Genome™, yang berisi ribuan "gen" yang secara otomatis ditugaskan untuk menggambarkan suasana hati, gaya, plot dan setting ke setiap film atau acara TV yang dirilis. Unsur-unsur ini kemudian disesuaikan dengan selera pribadi pelanggan sesuai dengan riwayat penayangannya untuk memberikan pengalaman penemuan yang benar-benar dipersonalisasi. Jinni juga memberikan rekomendasi, sesuai dengan favorit dan rating pengguna tertentu dari film dan acara TV. Rekomendasi didasarkan pada konten dan profil selera pengguna. Melalui situs Jinni dapat terhubung ke situs lain yang menyewakan atau menjual DVD atau menawarkan download atau streaming dengan biaya tertentu, seperti Netflix, Amazon dan Blockbuster. Produk Jinni termasuk situs web dan API untuk operator TV dan penyedia konten Internet. Mitra Jinni termasuk SeaChange, NDS, dan OpenTV.

2.5 Data Mining dalam Mesin Rekomendasi Film

Istilah data mining mengacu pada spektrum pemodelan matematika yang luas, teknik dan perangkat lunak yang digunakan untuk menemukan pola dalam data. Pembangunan model dalam istilah data mining pada konteks aplikasi recommender ini digunakan untuk membuat peraturan atau menyusun model rekomendasi dari kumpulan data yang besar. Sistem rekomendasi yang menggabungkan teknik data mining menggunakan pengetahuan yang dipelajari dari tindakan dan atribut pengguna. Sistem ini sering didasarkan pada pengembangan profil pengguna yang bisa bersifat persisten (berdasarkan Data historis "konsumsi" demografis atau item), bersifat sementara (berdasarkan tindakan Selama sesi saat ini), atau keduanya. Berbagai macam Algoritma yang meliputi clustering pengelompokan, klasifikasi, Association Rule, dan Grafik kemiripan seperti Horting.

Teknik klaster bekerja dengan mengidentifikasi kelompok konsumen yang tampak memiliki preferensi yang sama. Begitu kelompok diciptakan, rata-rata

pendapat dari konsumen lain di clusternya dapat digunakan untuk membuat prediksi untuk personal rekomendasi. Teknik klaster biasanya menghasilkan rekomendasi yang kurang personal dibandingkan metode lainnya, dan dalam beberapa kasus, Cluster memiliki akurasi yang buruk. Namun cluster terlihat baik jika digunakan sebagai tahap awal untuk mendeteksi kedekatan pengguna yang memiliki selera yang sama seperti pada metode collaborative filtering.

Classifiers atau klasifikasi adalah model komputasi umum untuk memasukkan ke item ke beberapa kategori/kelas. Masukannya mungkin merupakan vektor fitur untuk item yang diklasifikasikan atau data tentang hubungan antar item. Salah satu cara untuk membangun mesin rekomendasi menggunakan classifier adalah dengan menggunakan informasi tentang suatu produk dan profil pengguna sebagai input, dan untuk memiliki kategori output mewakili seberapa kuatnya merekomendasikan produk ke pelanggan. Klasifikasi dapat diimplementasikan dengan menggunakan banyak strategi mesin pembelajaran yang berbeda diantaranya rule induction, neural networks, dan Bayesian. Dalam setiap kasus, klasifikasi dilatih menggunakan pelatihan di mana klasifikasi aktual tersedia. Har tersebut diterapkan untuk mengklasifikasikan item baru yang kebenaran dasarnya tidak tersedia. Misalnya, jaringan Bayesian membuat model berdasarkan set pelatihan dengan Pohon keputusan di setiap simpul dan tepi yang mewakili informasi pengguna. Modelnya bisa Dibangun secara off-line selama hitungan jam atau hari. Model yang dihasilkan sangat kecil, sangat cepat, dan pada dasarnya sama akuratnya dengan matriks faktorisasi. Bayesian banyak digunakan sebagai fase kedua baik pada metode *collaborative filtering*, *content-based filtering* maupun pada *hybrid*.

Salah satu contoh data mining yang paling terkenal dalam pembuatan engine rekomendasi adalah Association rule, atau korelasi item-ke-item. Teknik ini mengidentifikasi item yang sering ditemukan di "asosiasi" dengan item. Asosiasi mungkin didasarkan pada co-purchase data, preferensi oleh pengguna umum, atau tindakan lainnya. Yang paling sederhana Implementasi, item-to-item korelasi dapat digunakan untuk mengidentifikasi "item yang cocok" untuk satu barang, seperti barang pakaian lainnya yang biasa dibeli dengan sepasang celana.

Horting adalah teknik berbasis grafik dimana node adalah pengguna, dan tepi antara node menunjukkan tingkat kesamaan antara dua pengguna. Prediksi

diproduksi dengan cara menjalankan grafik ke simpul terdekat dan menggabungkan pendapat dari pengguna terdekat.

Teks mining juga banyak digunakan untuk membuat mesin rekomendasi terutama dalam menggunakan metode content-based filtering dengan mencari kedekatan sebuah item yang disukai user berdasarkan fitur contentnya seperti judul, genre, aktor, dll. *Text mining* mencoba menemukan pengetahuan dari dokumen teks. Pengambilan term biasanya merupakan langkah awal dalam proses penambangan teks. Begitu term ditemukan, beberapa teknik penambangan teks lainnya dapat digunakan untuk meningkatkan *content-based filtering*. Dua teknik penambangan teks ini adalah pengelompokan dokumen/clusstering dan penggunaan thesauri.

Untuk menemukan item menarik, metode *content-based filtering* harus menelusuri seluruh koleksi item. Dengan mempartisi koleksi item ke dalam kelompok, ruang pencarian dapat dikurangi. Salah satu pendekatan penambangan teks adalah dengan menggunakan metode pengelompokan aglomeratif hirarkis untuk membuat kumpulan item terkait. Item diwakili sebagai vektor dalam model ruang vektor dan dibandingkan dengan menggunakan koefisien. Item yang diwakili dalam model ruang vektor berdasarkan pemilihan istilah tunggal tidak dapat sepenuhnya diidentifikasi. Ada sejumlah pendekatan *text mining* yang berkonsentrasi pada proses pemilihan istilah itu sendiri untuk memecahkan masalah sinonim dan polisemi. Proses parsing dapat diperluas untuk mengidentifikasi frase misalnya. Banyak sistem pencarian informasi mengidentifikasi frase sebagai pasangan istilah yang sering terjadi yang tidak dipisahkan. Metode yang lebih kompleks menggunakan algoritma dari pemrosesan bahasa alami untuk memilih frase.

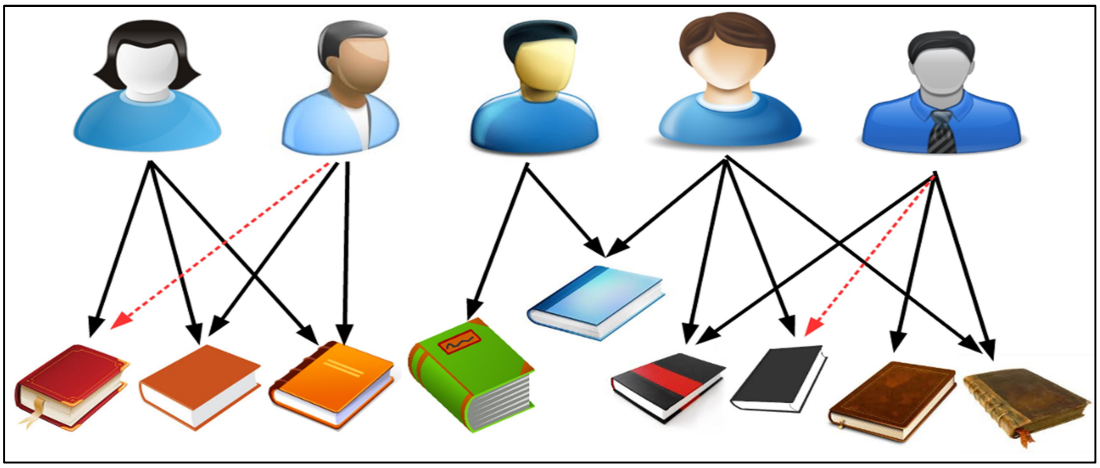
Pendekatan lain adalah mengelompokkan sinonim istilah bersama dengan menggunakan tesaurus. Tesaurus adalah seperangkat istilah ditambah serangkaian hubungan antara istilah-istilah ini. Tesaurus bisa dihasilkan baik secara manual maupun otomatis. Sebuah tesaurus buatan tangan biasanya hanya berisi pengetahuan khusus domain karena konstruksinya adalah proses yang sangat padat karya. Konstruksi otomatis tesaurus memerlukan pengelompokkan istilah yang sering terjadi bersamaan dalam pengumpulan item. Salah satu pendekatannya adalah untuk mewakili istilah sebagai vektor, di mana setiap bobot sesuai dengan jumlah kejadian dari istilah dalam dokumen tertentu. Istilahnya dibandingkan dengan

menggunakan ukuran kosinus. Berbagai metode pengelompokan kemudian dapat digunakan untuk menemukan kelompok istilah yang sering terjadi bersamaan.

Dari beberapa jenis algoritma yang digunakan dalam *data mining* dan teks mining yang digunakan untuk membuat mesin rekomendasi, kelima jenis algoritma tersebut dapat dimasukkan dan digunakan untuk ketiga metode mesin rekomendasi yaitu *collaborative filtering, content-based filtering dan Hybrid. Collaborative filtering* banyak menggunakan data mining dengan algoritma clustering, horting dan assosiation rule, sementara *content-based filtering* banyak menggunakan teknik *text mining* dengan algoritma kemiripan teks berdasarkan vektor atau jarak terdekat dokumen dengan *query*, dan *Hybrid* yaitu penggabungan *collaborative filtering* dan *content-based filtering* banyak menggunakan *bayesian classifier* atau *neural network*.

2.6 Collaborative Filtering

Ide utama dalam *collaborative filtering* adalah untuk memanfaatkan opini atau penilaian pengguna lain yang ada untuk memprediksi item yang mungkin akan disukai/diminati oleh seorang pengguna [10]. Kualitas rekomendasi yang diberikan dengan menggunakan metode ini sangat bergantung dari peniliaian pengguna lain terhadap suatu item. Seperti dikemukakan di bagian penjelasan sistem rekomendasi bahwa penilaian dapat berbentuk *explicit* maupun *implicit* dimana Pendekatan *implicit*, artinya, sistem menyimpan dan mempelajari perilaku pengguna terhadap item, contohnya item apa yang pernah dibeli pengguna, berapa kali penggula melihat barang tersebut, dsb. Sementara pendekatan *explicit*, yaitu dengan menanyakan kepada pengguna secara langsung deskripsi item yang bagaimana yang ia sukai/minati contoh keluarannya berupa rating atau kuesioner [23].



Gambar 2.3. Konsep *collaborative filtering* [25]

Dari Gambar 2.1 terlihat konsep dari *collaborative filtering* dalam mengolah profil pengguna didapatkan dari pengolahan profil pengguna terhadap ketertarikan terhadap suatu item, dua orang yang mempunyai ketertarikan yang sama dianggap memiliki selera yang sama terhadap suatu item, oleh karena itu item yang kemudian disukai oleh orang tersebut di rekomendasikan kepada orang yang memiliki selera yang sama dengan harapan orang tersebut dapat memiliki ketertarikan mengenai item tersebut sama dengan orang yang memiliki selera yang sama.

Terdapat dua tahapan proses yang dilakukan pada teknik *collaborative filtering* dalam membuat rekomendasi, yaitu :

1. Prediksi

Prediksi opini akan diberikan oleh sistem. Pada tahapan ini dilakukan pelatihan (training, dengan tujuan untuk memperoleh suatu model yang paling optimal dalam artian memperoleh akurasi dengan nilai kesalahan (*error*) yang paling kecil

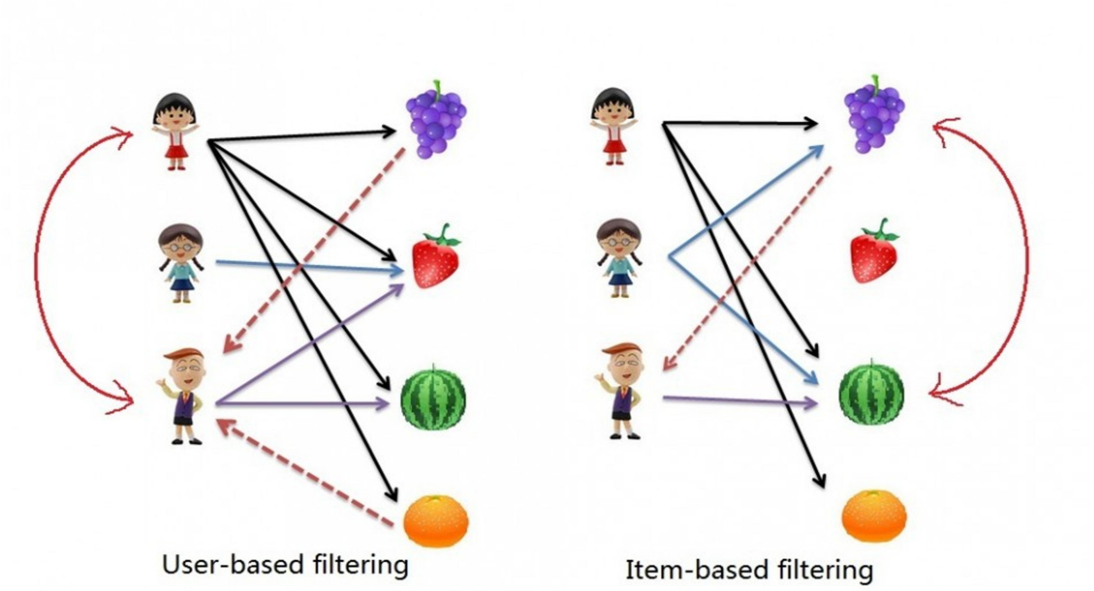
2. Rekomendasi

Memberikan rekomendasi berupa daftar produk dengan nilai prediksi tertinggi yang mungkin akan disukai pengguna. Hal ini sering disebut Top N recommendation dan personal recommendation. Pada tahap ini setelah mendapatkan model yang paling optimal, model tersebut diujikan pada data penguji dengan tujuan untuk mendapatkan hasil peringkat sebagai output dari pemilihan data penguji tersebut.

Untuk mengimplementasikan collaborative filtering terdapat dua jenis pendekatan yaitu yang berbasis memori (*memory-based collaborative filtering*) dan pendekatan berbasis model (*model-based collaborative filtering*).

2.6.1 Memory-Based Filtering

Dalam pendekatan berbasis memori hal utama yang dilakukan adalah menggunakan data rating untuk dihitung kemiripannya (*simalirity*). Kemiripan ini mungkin terdapat pada pengguna ataupun pada produk. Berdasarkan nilai yang diperoloh inilah kemudian yang digunakan untuk menentukan nilai prediksi atau rekomendasi. Terdapat dua jenis pendekatan yang digunakan dalam *memory-based collaborative filtering* yaitu *user based filtering* dan *item based filtering* seperti pada Gambar 2.2

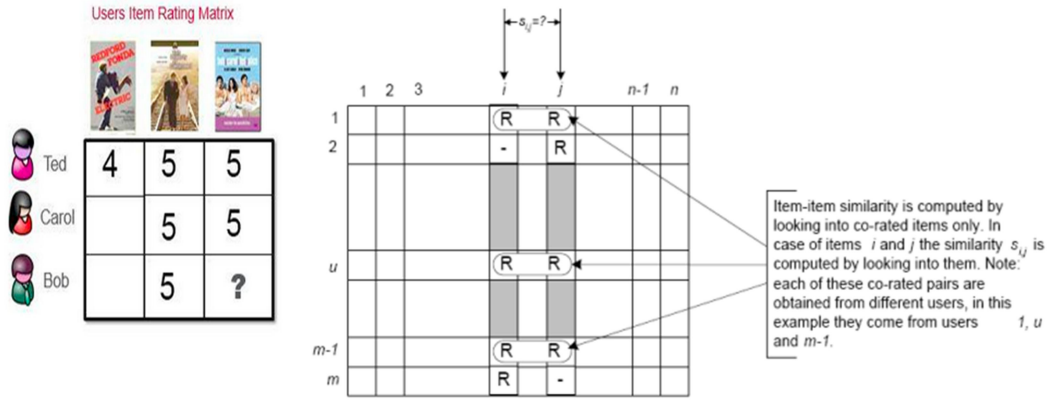


Gambar 2.4. Konsep perbedaan user-based filltering dan item based filtering [26]

User based collaborative filtering bekerja berdasarkan asumsi bahwa setiap pengguna merupakan bagian dari kelompok yang memiliki kesamaan dengan pengguna lainnya [24]. Dengan kata lain, pengguna yang memiliki kesamaan hubungan (atribut) akan tertarik terhadap item yang sama.

Item-Based collaborative filtering adalah algoritma yang bekerja untuk mencari hubungan antar item berdasarkan tabel rating untuk membentuk sebuah rekomendasi terhadap suatu item kepada user. *Item-based collaborative filtering* berasumsi bahwa jika mayoritas pengguna memberi penilaian beberapa item secara

serupa, pengguna yang kita targetkan juga akan memberi penilaian terhadap item-item tersebut secara serupa dengan mayoritas pengguna lain. Ilustrasi collaborative filtering yang mengolah rating sebagai acuan dapat dilihat pada Gambar 2.3



Gambar 2.5. Ilustrasi *item-based collaborative filtering* dan *matriks item –based collaborative filtering* [27]

Salah satu cara untuk menghitung kemiripan adalah dengan menghitung nilai kemiripan diantara item yang telah dirating oleh pengguna. Untuk membuat nilai kemiripan, digunakan persamaan *adjusted-cosine similarity* seperti dalam persamaan (1)

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}u)(R_{u,j} - \bar{R}u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}u)^2}} \quad (1)$$

Dimana:

Sim (i,j) = Nilai kemiripan antara item i dan item j

$u \in U$ = Himpunan pengguna u yang merating item i dan item j

$R_{u,i}$ = Rating pengguna u pada item i

$R_{u,j}$ = Rating pengguna u pada item j

$\bar{R}_u$ = Nilai rata-rata rating pengguna u

Sesudah nilai kemiripan antar item didapatkan, maka tahap selanjutnya adalah membuat prediksi rating terhadap item yang belum dirating oleh pengguna. Untuk menghitung prediksi rating, digunakan persamaan *weighted sum* dengan

menerapkan *nearest neighbor* yaitu menentukan jumlah *neighbor* yang digunakan dalam proses penghitung prediksi.

$$P(u,j) = \frac{\sum_{i \in I} {}^{K}_{u}(j)\left(R_{u,i} * S_{i,j}\right)}{\sum_{i \in I} {}^{K}_{u}(j)\left|S_{i,j}\right|} \quad (2)$$

Dimana :
P(u,j) = Prediksi untuk pengguna u pada item j

$I \in I^{K}_{u}(j)$ = Himpunan K item yang mirip dengan item j
$R_{u,i}$ = Rating pengguna u pada item i
$S_{i,j}$ = Nilai kemiripan antara item i dan item j

Dalam proses memprediksi suatu rating item terhadap seorang pengguna, sebelumnya ditentukan dahulu berapa jumlah K yang akan digunakan. K dalam hal ini merupakan sejumlah item yang telah dirating oleh pengguna dan mempunyai nilai kemiripan paling tinggi dengan item yang akan diprediksi ratingnya.

Setelah model diperoleh dilakukan pengujian keakurasian dengan menggunakan data penguji sehingga didapat nilai keakurasian yang paling optimum. Adapun persamaan yang digunakan untuk mengetahui tingkat keakurasiannya adalah dengan menggunakan persamaan root mean squared error (RMSE).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Puj - Rui)^2} \quad (3)$$

Dimana :
n = banyaknya entri yang dirating oleh pengguna
Puj= rating prediksi dari pengguna u terhadap produk i pada data penguji
Rui= rating sebenarnya yang diberikan pengguna u pada produk i

Berdasarkan penjelasan diatas maka dapat disimpulkan bahwa kelebihan dari *memory-based* adalah sangat sederhana dan mudah untuk diterapkan, namun memiliki kekurangan diantaranya adalah skalabilitas, sparsitas dan *overfiting*. Skalabilitas dalam *memory-based* menjadi kelemahan dari *model-based* karena membutuhkan memori komputer yang besar karena memori diperlukan untuk proses seluruh database pada saat membuat prediksi. Sparsitas juga menjadi kelemahan dari memori based yaitu pada matriks user item yang besar, tidak semua terisi penuh,

pengguna hanya merating beberapa item akibatnya terdapat matriks yang jarang (*sparse*), hal tersebut mengakibatkan ada beberapa pengguna aktif tidak mendapatkan prediksi karena pengguna aktif tidak memiliki barang yang sama dengan semua orang yang telah memberi nilai pada item target. *Overfiting*, dibutuhkan semua variabilitas acak dalam penilaian orang sebagai sebab akibat dan tidak ada pengujian. Dengan kata lain, algoritma berbasis memori tidak menggeneralisasi data sama sekali.

2.6.2 Model Based Filtering

Untuk memperbaiki kelemahan dari *memory-based collaborative filtering* salah satu alternatifnya menggunakan *model-based collaborative filtering***.** Berbeda dengan pendekatan yang berbasis memori (*memory-based collaborative filtering*), dalam pendekatan berbasis model (*model-based collaborative filtering*) diperlukan contoh data pelatihan (training) untuk memprediksi produk-produk yang terdapat pada data penguji (testing). Pendekatan berbasis model mampu memprediksi nilai untuk produk-produk yang tidak pernah dirating oleh pengguna pada data pelatihan, model akan diperoleh dengan menetapkan metode ini pada data pelatihan, selanjutnya model digunakan untuk memprediksi produk-produk yang terdapat pada data penguji. Untuk pembuatan rekomendasi dapat digunakan faktorisasi matriks atau *LU-decomposition*.

Faktorisasi matriks adalah metode yang memfaktorkan sebuah matriks, misalnya untuk mencari dua (atau lebih) matriks-matriks, sehingga ketika dikalikan akan didapatkan kembali matriks yang nilainya sama atau mendekati nilai matriks aslinya. faktorisasi matriks mengkarakterkan antara user dan item dengan vektor dari faktor yang muncul dari pola rating user. Faktor keterkaitan yang besar antara user dan item mengarahkan pada sebuah rekomendasi. Metode ini menjadi populer dengan mengombinasikan perkiraan yang baik dengan akurasi prediksi. Faktorisasi matriks atau *LU-decomposition* dapat digunakan untuk menemukan latent factor/latent features berdasarkan interaksi antara dua macam entitas yang berbeda, dan untuk memprediksi rating dalam item. Langkah pertama adalah dengan mengasumsikan bahwa sistem persamaan linier dapat diasumsikan dengan persamaan matriks

$$Qx = r \quad (4)$$

Pada metode LU-decomposition, matrik Q difaktorkan menjadi matrik U (user) dan matrik I(item), dimana dimensi atau ukuran matrik U dan I harus sama dengan dimensi matrik Q. Atau dengan kata lain, hasil perkalian matrik U dan matrik I adalah matrik Q,

$$Q = UI \quad (5)$$

sehingga persamaan (4) menjadi

$$UIx = r \quad (6)$$

Langkah penyelesaian sistem persamaan linear dengan metode LU-decomposition, diawali dengan menghadirkan vektor y dimana,

$$Ux = y \quad (7)$$

Langkah tersebut tidak bermaksud untuk menghitung vektor y, melainkan untuk menghitung vektor x. Artinya, sebelum persamaan (7) dieksekusi, nilai-nilai yang menempati elemen-elemen vektor y harus sudah diketahui. Untuk memperoleh nilai vektor y adalah :

$$Iy = r \quad (8)$$

Kesimpulannya, metode *LU-decomposition* dilakukan dengan tiga langkah sebagai berikut:

- Melakukan faktorisasi matrik Q menjadi matrik U dan matrik I → Q=UI.
- Menghitung vektor y dengan operasi matrik Uy = r. Ini adalah proses forward substitution atau substitusi-maju.
- Menghitung vektor x dengan operasi matrik Ux = y. Ini adalah proses backwardsubstitution atau substitusi-mundur.

Metode *Non – Negative Matrix Factorization* (NMF) menyerupai dengan metode *Matrix Factorization*, akan tetapi nilai semua elemen pada NMF ini memiliki sebuah batasan yaitu semua nilai tersebut tidak dapat bernilai kurang dari 0. Ini bertujuan untuk menghilangkan pengaruh interaksi elemen User dan Item yang bernilai negatif.

A. *Singular Value Decomposition* (SVD)

Setiap proses dekomposisi akan memfaktorkan sebuah matriks menjadi lebih dari satu matriks. *Singular Value Decomposition* (SVD) berkaitan erat dengan

singular value atau nilai singular dari sebuah matriks yang merupakan salah satu karakteristik matriks. Bentuk dekomposisi SVD sebagaimana dalam persamaan (9)

$$A=USV^T \quad (9)$$

Dimana :

A = matriks *mxn*

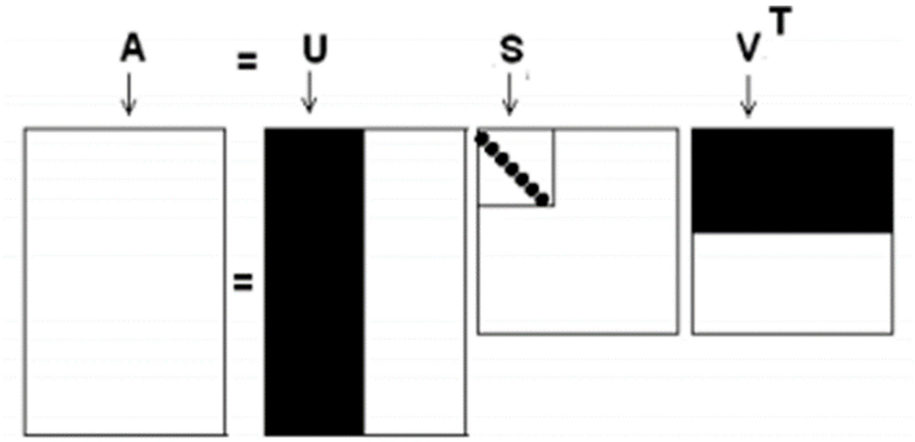
U = Matriks yang dibentuk oleh eigen vektor normal matriks AA^T

S = Matriks singular yatu matriks diagonal *mxn* yang entri-entrinya adalah nilai singular dari matriksk A yang elemen diagonalnya terurut turun dan non-negatif

V = Matriks yang dibentuk oleh eigen vektor normal matriks A^TA

V^T=Transpose dari matriks V

Bila digambarkan maka satu matrik utuh A akan didekomposisi menjadi 3 matrik seperti Gambar 4



Gambar 2.6. Dekomposisi 1 matriks menjadi 3 matriks

Dalam perhitungan SVD pertama-tama kita perlu menghitung nilai eigen (nilai eigen merupakan nilai yang mempresentasikan suatu matriks dalam perkalian dengan suatu vektor) dan vektor eigen (merupakan solusi dari matriks untuk setiap nilai yang ada)dari$AA^T \ dan \ A^TA$ Vektor eigen dari AA^T bentuk kolom dari U, sedangkan vektor eigen dari A^TASuatu bentuk kolom V. Selain itu, nilai-nilai singular (SVs) di S adalah akar kuadrat dari nilai eigen dari A^TA atau AA^T. Kelemahan dari SVD adalah :

- Kompleksitas komputasinya

- Fakta bahwa matriks asli harus sangat padat (tidak ada nilai yang hilang), dan seringkali kasus di mana matriks yang harus didekomposisi jarang (memperkenalkan nilai yang hilang).

B. *Stocastic Gradient Decent*

Gradient Descent adalah algoritma optimasi orde pertama yang banyak digunakan di bidang pembelajaran mesin. *Stocastic Gradient Decent* (sering disingkat dalam SGD) merupakan pendekatan stokastik metode optimasi persentase gradien untuk meminimalkan fungsi objektif yang ditulis sebagai jumlah fungsi yang dapat didiferensiasi. SGD mencoba menemukan minima atau maxima dengan iterasi. Error pada fungsi objektif tertentu diminimalisir pada setiap iterasi dengan persamaan (10)

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Qi(w) \qquad (10)$$

Dimana :

Q(w) : Fungsi objektif dengan parameter w

Qi(w) : Nilai evaluasi pada fungsi objektif Q (w), iterasi ke-i

n : jumlah iterasi

Secara konseptual, SGD adalah algoritma optimasi secara iteratif sederhana yang mengasumsikan adanya fungsi biaya (*cost function*) dan nilai awal yang sewenang-wenang untuk variabel pengoptimalan. Namun, mengoptimalkan fungsi biaya dengan kemiringan gradien hanya menjamin konvergensi ke minima lokal. SGD bukan pilihan populer untuk pengoptimal faktorisasi matriks jika dimensi matriks rating sangat besar. Dengan kata lain, di SGD akan berulang kali memilih beberapa subset fungsi kerugian untuk meminimalkan - satu atau lebih

sel dalam matriks rating - dan menetapkan parameter 0 untuk membuat prediksi lebih baik.

C. *Alternating Least Square – Weight Regularization*

Beberapa algoritma menggunakan matriks faktorisasi seperti disebutkan diatas dapat menyelesaikan matriks faktorisasi namun kelemahan masih ada terutama pada masalah skalanilitas dan sparsitas, namun yang terbaru dan memberikan hasil cukup baik diantaranya adalah *Alternating Least Square – Weight Regularization*

(ALS-WR). Penelitian sebelumnya mengenai ini [16] [17] [18] [19] membuktikan bahwa ALS-WR memiliki performance yang lebih baik dibandingkan dengan kedua algoritma sebelumnya. Oleh karena itu peneliti menggunakan algoritma ALS-WR untuk proses collaborative filtering.

Konsep dasar ALS adalah matriks besar dari interaksi pengguna dan item dan mengetahui fitur laten (atau disembunyikan) yang berhubungan antara user dan item dan itu direduce dalam matriks jauh lebih kecil. Itulah yang ALS coba lakukan melalui matriks faktorisasi [24]

Matriks preferensi R diasumsikan dapat difaktorisasi (dipecah) menjadi 2 matriks yang lebih kecil. Jika dimensi matriks preferensi = X × Y,dimana X adalah user dan Y adalah Item maka hasil faktorisasinya adalah matriks X dengan sebutan $X \in \mathbb{R}^{m\,x\,f}$b dan matriks Y dengan $Y \in \mathbb{R}^{f\,x\,n}$ dimana f adalah rank atau faktor laten dari faktorisasi ini yang mempengaruhi X dan Y. Hasil kali baris X dan kolom Y menyatakan kesesuaian pengguna X dengan konte.

Namun, pada kebanyakan database *e-commerce* maupun rental film, beberapa user cendrung hanya memberikan rating pada item yang sudah dibelinya atau dilihatnya, akibatnya matriks Q tidak seluruhnya terisi, disini ALS-WR bekerja yaitu memprediksi faktor laten tersebut dengan W atau weight untuk identifikasi dengan cara :

$$w_{ui}=\begin{cases}0 & if\ q_{ui}=0\\ 1 & else\end{cases} \tag{11}$$

Karena pada prinsipnya algoritma ini hanya memprediksi nilai untuk item yang telah dinilai oleh pengguna sebelumnya, tidak pada item yang belum di nilai sama sekali, sesuai dengan prinsip *collaborative filtering* adalah *wisdom crowd* atau penilaian bergantung pada penilaian publik, seperti sistem pemasaran *word to mouth* yang telah dijelaskan pada bagian pendahulan. Amazon, yahoo maupun netflix mempunyai trik tersendiri untuk mengatasi masalah item baru (*cold start problem*) yaitu dengan menampilkan item baru di halaman depan web.

Lebih lanjut, selain memiliki *weight*, algoritma ini juga memiliki *lamda* (λ) yang berfungsi sebagai paramater regulasi agar rekomendasi yang dihasilkan mengarah

kepada item yang bernilai baik sesuai dengan preferensi pengguna . Maka dari penjelasan dan logika diatas didapatkan fungsi (12) dan (13)

$$J(x_u) = (q_u - x_u Y) W_u (q_u - x_u Y)^T + \lambda x_u x_u^T \quad (12)$$

$$J(x_{yi}) = (q_i - X y_i) W_i (q_i - X y_i)^T + \lambda y_i y_i^T \quad (13)$$

Untuk mendapatkan hasil dari persamaan diatas, maka paramater regulasi harus disetel menggunakan cross-validasi agar algoritma dapat digenerate lebih baik. Solusinya untuk nilai yang hilang adalah dengan menyelesaikan fungsi(14) dan (15)

$$x_u = (Y W_U Y^T + \lambda I)^{-1} Y W_U q_u \quad (14)$$

$$x_u = (X^T W_i X + \lambda I)^{-1} X^T W_i q_i \quad (15)$$

Melalui Iterasi pada mesin pembelajaran, proses tersebut dilakukan berulang-ulang hingga mencapai error terkecil atau titik konvergen. Hasil dari Iterasi kemudian menjad model yang akan digunakan untuk memprediksi data yang hilang, kemudia rating dapat diurutkan untuk menghasilkan Top N yang menjadi dasar pembuatan rekomendasi personal. Pada pembuatan rekomendasi personal (*user based*), rating baru dari user baru ditambahkan dan dimasukkan kedalam proses perhitungan dengan rumus (12) dan (13) untuk kemudian diprediksi ulang dengan error terkecil dan menghasilkan TopN untuk user baru tersebut (personal recommendation).

2.7 *Content-Based Filtering*

Mesin rekomendasi *content-based filtering* menggunakan ketersediaan konten (sering juga disebut dengan fitur, atribut atau karakteristik) sebuah item sebagai basis dalam pemberian rekomendasi [10]. Sebagai contoh, sebuah film mempunyai konten seperti judul film, genre, author, tahun rilis, dan lain-lain, atau sebuah file dokumen memiliki konten berupa tulisan yang ada di dalamnya.

Sistem rekomendasi dengan metode *content based filtering* biasa digunakan untuk merekomendasikan berita, artikel maupun situs web. Metode tersebut akan mengekstrak informasi yang terdapat pada item kemudian membandingkannya dengan informasi item yang pernah dilihat atau disukai oleh user. Sistem rekomendasi berbasis konten memiliki beberapa kelebihan, yaitu :

- Sistem rekomendasi berbasis konten disusun berdasarkan fitur content yang pernah ditelusuri atau dipilih pengguna
- Sistem rekomendasi berbasis konten dapat merekomendasikan item-item yang bahkan belum pernah di-rate oleh siapapun.

Namun, sistem rekomendasi berbasis konten juga memiliki beberapa kelemahan, yaitu Sistem rekomendasi berbasis konten tidak memiliki kemampuan untuk dapat memeberikan hasil rekomendasi yang tidak terduga.

2.7.1 Kemiripan Genre

Konsep Kemiripan Genre sebenarnya mengambil konsep dari content-based filtering yaitu menggunakan fitur kontent berupa genre sebagai filter untuk menghasilkan rekomendasi. Seperti dikemukakan diatas, mesin rekomendasi *content-based filtering* menggunakan ketersediaan konten. Salah satu konten yang paling banyak dijadikan rujukan adalah judul dan topik/tema. Judul melekat pada item itu sendiri sementara topik/tema/genre atau klasifikasi kontent diberikan oleh seseorang yang ahli dibidang item tersebut sehingga dapat memutuskan item tersebut masuk kedalam klasifikasi tertentu. Biasanya *content based filtering* digunakan untuk sistem rekomendasi item tekstual seperti buku atau berita. Pada item film konten yang paling sering digunakan adalah artis, judul dan genre. Untuk artis dan title mungkin tidak memberikan hasil yang mengejudkan karena dapat dikatan hasilnya akan sama seperti pada mesin pencari, tetapi untuk genre pengguna bisa mendapatkan film berbagai macam judul dengan genre yang sama seperti yang pernah dipilih atau ditonton.

Ada beberapa algoritma untuk menghasilkan rekomendasi berdasarkan genre. Disini penulis memilih *cosine similiarity*. Pemilihan ini didasarkan tingkat presisi yang tinggi dihasilkan oleh algoritma ini. Cara kerja algoritma ini adalah kemiripan antara suatu *query* (Q) dengan daftar item (dengan semua dokumen). Kemudian dilakukan pengurutan dan dikembalikan kepada pengguna. Kemiripan antar dokumen *cosine similarity* menggunakan rumus (16)

$$Sim\left(\underset{d_j}{\rightarrow}.\underset{q}{\rightarrow}\right)=\frac{\sum_{i=1}^{t}(w_{ij}\ x\ w_{iq})}{\sqrt{\sum_{i=1}^{t}(w_{ij})^2}\ x\ \sqrt{\sum_{i=1}^{t}(w_{iq})^2}} \quad (16)$$

Dimana:

$\underset{d_j}{\rightarrow}$: vektor item ke-j

$\underset{q}{\rightarrow}$: vektor kata kunci

w_{ij} : bobot *index term* i pada item j

w_{iq} : bobot *index term* I pada kata kunci q

Langkah dalam menghitung rumus *cosine similarity* adalah (a) hitung hasil perkalian scalar antara Q dan data uji, hasilnya perkalian dari setiap dokumen dengan Q dijumlahkan (sesuai pembilang pada rumus di atas). (b)hitung panjang setiap dokumen, termasuk Q. Caranya dengan mengkuadratkan bobot setiap term dalam setiap dokumen, jumlahkan nilai kuadrat dan terakhir di akar kan. Untuk menghitung bobot digunakan *Term Frequency- Inverse Document Frequency* (Tf-Idf) dengan tahapan penghitungan pada (17), (18) dan (19).

1. menghitung *term frequency* (tf) :

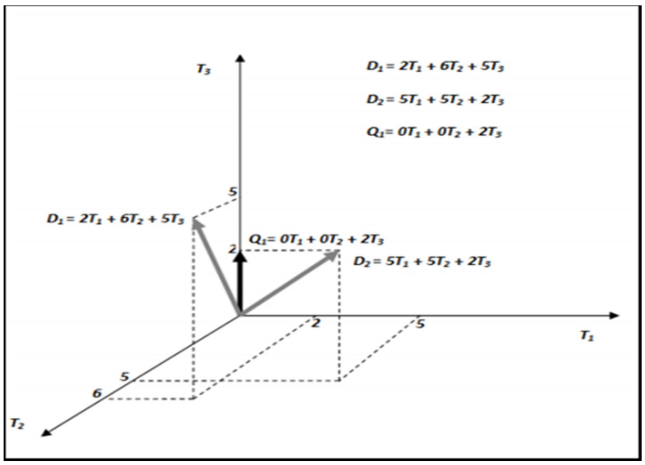
$$tf = tf_{ij} \quad (17)$$

2. menghitung *Inverse Document Frequency* (idf) :

$$idfi = \log (N/df_j) \quad (18)$$

3. menghitung bobot tiap *term :*

$$Wij = tfij.\log\frac{N}{dfi} \quad (19)$$



Gambar 2.7 Contoh Model Ruang Vektor

Jika dua dokumen D1 = 2T1 + 6T2 + 5T3 dan D2 = 5T1 + 5T2 + 2T3 dan keyword Q1 = 0T1 + 0T2 + 2T3 sebagaimana diperlihatkan pada Gambar 2.8, berikut ini adalah nilai kosinus yang diperoleh:

$$\text{Sim}\left(\underset{d_1}{\rightarrow}.\underset{q}{\rightarrow}\right) = \frac{(2x0) + (6x0) + (5x2)}{\left(\sqrt{4+36+25}\right)\left(\sqrt{0+0+4}\right)} = \frac{10}{\sqrt{65,4}} = 0,62$$

$$\text{Sim}\left(\underset{d_2}{\rightarrow}.\underset{q}{\rightarrow}\right) = \frac{(5x0) + (5x0) + (2x2)}{\left(\sqrt{25+25+0}\right)\left(\sqrt{0+0+4}\right)} = \frac{4}{\sqrt{54,4}} = 0,27$$

Contoh di atas memperlihatkan bahwa sesuai dengan perhitungan kosinus, dokumen D2 lebih mirip dengan keyword daripada dokumen D1. Terlihat sudut antara D2 dan Q1 lebih kecil dari pada sudut antara D1 dan Q1.

Pada penelitian ini *query* yang dituju adalah genre item yang di klik oleh pengguna, misalnya pengguna mengklik film *American Beauty* yang bergenre *Romance* and *commedy*, maka sistem akan mencari film dengan genre serupa lalu dilakukan pembobotan menggunakan Tf-Idf. Hasil dari pembobotan kemudian dilakukan perhitungan kemiripan dengamemvektorkan antara dokumen dengan query, nilai cosine dihitung dari sudut kedua vektor antara dokumen dengan query.

2.8 Penggabungan *content-based* dan *collaboration filtering*

a. Hybrid

Hybrid menurut Burke dalam artikelnya yang berjudul Hybrid Web Recommender Systems [28] digunakan untuk menggambarkan setiap sistem rekomendasi yang menggabungkan beberapa teknik rekomendasi untuk menghasilkan sebuah output. Lebih lanjut, Burke kemudian dibagi kembali menjadi 7 kategori metode sebagai berikut :

1. Weighted hybrid : Nilai komponen dari sistem rekomendasi yang berbeda digabungkan secara numerik atau menggunakan algoritma linier.
2. Switching hybrid : Sistem memilih komponen-komponen dari setiap rekomendasi dan menerapkan komponen yang dipilih.
3. Mixed hybrid : Rekomendasi dari berbagai sistem rekomendasi disajikan bersama
4. Feature Combination : Fitur-fitur yang berasal dari berbagai sumber pengetahuan digabungkan dan diberikan algoritma rekomendasi
5. Feature Augmentation : merupakan salah satu teknik rekomendasi yang digunakan untuk menghitung sebuah fitur atau sekumpulan fitur yang kemudian menjadi bagian yang dimasukkan ke teknik berikutnya.
6. Cascade : merupakan rekomendasi yang memiliki prioritas tinggi sebagai solusi pemecahan masalah dalam melakukan perbaikan
7. Meta-level : merupakan salah satu teknik rekomendasi yang diterapkan dan menghasilkan beberapa jenis model, yang kemudian digunakan sebagai input oleh teknik berikutnya.

b. Content-Boosted Collaborative Filtering (CBCF)

Content-Boosted Collaborative Filtering yang merupakan recommender system yang menggabungkan antara content based filtering dengan collobarative filtering. Penggabungan pendekatan content based filtering dengan collaborative filtering pada metode CBCF bertujuan untuk menanggulangi kekurangan yang ada pada kedua pendekatan sebelumnya terutama First-Rater Problem dan Sparsity. Metode ini menggunakan content-based predictor untuk meningkatkan

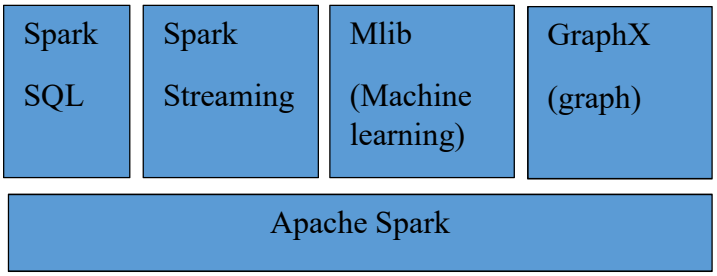
data user yang ada, dan kemudian memberikan rekomendasi melalui collaborative filtering. [29] [30]

c. Content Features Similiarities Based on Collaborative Filtering

Kebalikan dari metode CBCF, dalam metode ini menjadikan *collaborative filtering* sebagai landasan atau inputan ke teknik selanjutnya [15] [31] [32]. Tahap pertama yang harus diselesaikan adalah *collaborative filtering*, kemdian hasil dari *collaborative filtering* didekatkan kembali berdasarkan korelasinya atau kemiripannya berdasarkan *content features* yaitu berdasarkan content features yaitu *genre*, aktor, judul film, dll. Untuk mendekatkan berdasarkan kemiripannya dapat digunakan algoritma kemiripan (*cosine similiarity*), korelasi *(pearson corellation*) atau algoritma probablistik (*naive bayes classifier*)

2.9 Teknologi Mesin Pembelajaran BigData untuk Mesin Rekomendasi

Apache Spark dikembangkan di UC Berkeley AMPLab pada tahun 2009 dan bersumber pada tahun 2010, Apache Spark adalah teknologi pengolahan *big data* yang dirancang untuk Batch dan streaming beban kerja dalam waktu singkat. Apache Spark adalah *Open Source* analisa data klastering frame work. Spark berada di komunitas Hadoop Open Source. Spark dibangun diatas Hadoop Distributed File System (HDFS).



Gambar 2.8Arsitektur Spark

Walaupun Spark dan Hadoop sama-sama eksis, Spark melakukan lebih baik daripada Hadoop dalam hal aplikasi spesifik tertentu. Spark tidak dibatasi oleh dua tahap paradigma *mapreduce*. Ini memberikan 100 kali kinerja yang lebih baik daripada Hadoop untuk aplikasi tertentu [36]. Spark menyediakan kemampuan untuk dalam komputasi klustering yang memungkinkan pengguna untuk memproses data ke dalam memori *cluster*. Data yang diproses ke dalam memori utama dapat

digunakan berulang kali dan ini mempercepat seluruh waktu respon. Properti ini membuat Spark cocok untuk algoritmamesin pembelajaran.

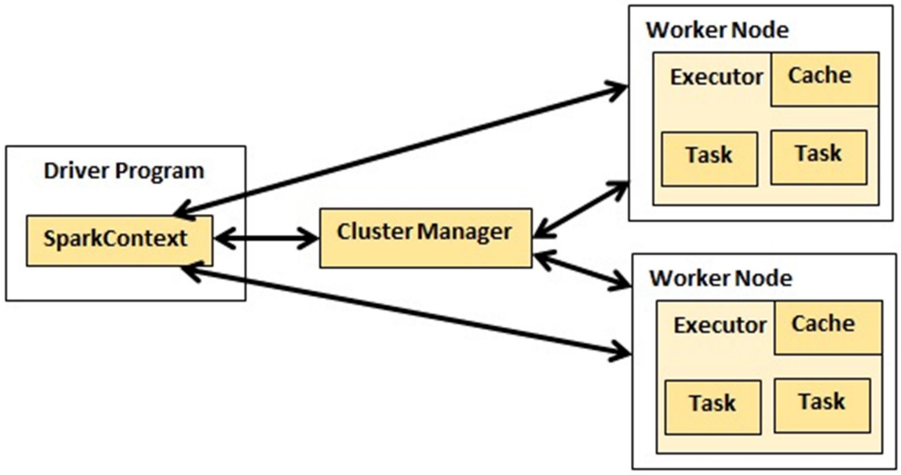
Apache Spark menyediakan aplikasi antarmuka pemrograman yang berpusat pada struktur data yang disebut resilient distributed datasets (RDD), *read-only multiset item* data didistribusikan ke sekelompok mesin, yang dipertahankan dalam cara toleran . Hal ini dikembangkan dalam solusi keterbatasan dalam Mapreduce , sebuah read-only multiset item data didistribusikan ke sekelompok mesin, yang dipertahankan dengan cara fault tolerant. Hal ini dikembangkan dalam menanggapi keterbatasan dalam paradigma komputasi Mapreduce cluster, yang memaksa struktur linear dataflow tertentu di program yang didistribusikan: program Mapreduce membaca data masukan dari disk, memetakan fungsi seluruh data, menurunkan hasil peta, dan mengurangi hasil penyimpanan di dalam disk. Fungsi RDDS pada Spark sebagai working set program terdistribusi yang menawarkan (dengan sengaja) formulir dari transfer memori yang terdistribusi.

MLlib adalah kerangka mesin pembelajaran terdistribusi di atas Spark Core, karena sebagian besar dari arsitektur Spark berbasis memori terdistribusi, terukur sembilan kali lebih cepat dalam implementasi berbasis disk dibanding Apache Mahout (menurut pengukuran yang dilakukan oleh pengembang MLlib terhadap implementasi Alternating Least Squares (ALS), dan sebelum Mahout sendiri memperoleh tampilan Spark), dan skala yang lebih baik daripada Vowpal Wabbit. Banyak mesin pembelajaran yang biasa digunakan dan algoritma statistika telah dilaksanakan dan diselesaikan dengan MLlib yang menyederhanakan pipa skala besar mesin pembelajaran, termasuk:

- Ringkasan perhitungan statistika*, correlations, stratified sampling, hypothesis testing, random data generation*
- *classification and regression: support vector machines, logistic regression, linear regression, decision trees, naive Bayes classification*
- c*ollaborative filtering* techniques termasuk *alternating least squares* (ALS)
- *cluster analysis methods k-means,* dan *Latent Dirichlet Allocation (LDA)*
- d*imensionality reduction techniques* seperti *singular value decomposition (SVD)*, dan *principal component analysis(PCA)*

- Ekstraksi fitur dan fungsi-fungsi transformasi
- *optimization algorithms stochastic gradient descent, limited-memory BFGS (L-BFGS)*

Desain dan cara kerja MLlib yang sederhana memungkinkan menjalankan berbagai algoritma pada dataset didistribusikan (RDD), mewakili semua data sebagai RDDS. Setiap dataset di RDD dibagi menjadi partisi logis, yang dapat dihitung pada node yang berbeda dari cluster. RDDS dapat berisi jenis Python, Java, atau Scala, termasuk kelas yang ditetapkan pengguna. Untuk menggunakan Mlib dimulai dengan RDD string yang mewakili pesan, kemudian jalankan algoritma yang terdapat dalam fitur Mlib, setelah mendapatkan hasil kemudian evaluasi model pada dataset uji menggunakan salah satu fungsi evaluasi/validasi Mlib ini. Sementara dalam mesin learning ini terdapat proses seperti pada Gambar 2.8



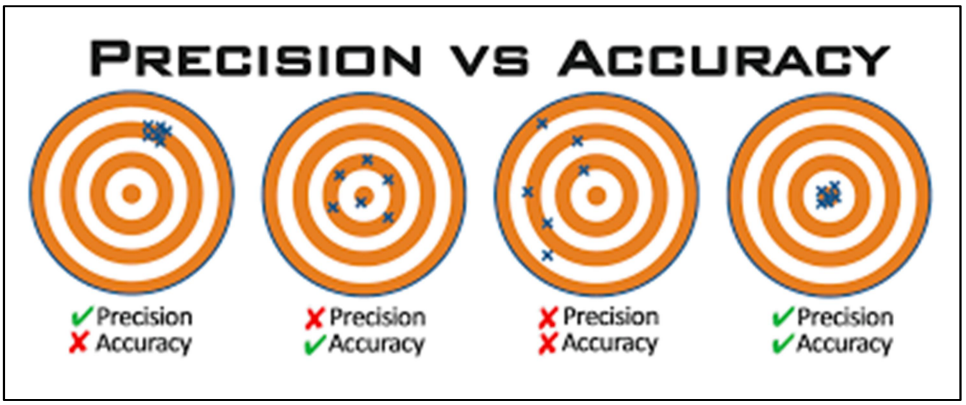
Gambar 2.9.Cluster Process on Mlib Spark [36]

Dimana dalam menjalankan algoritmanya, spark mendistribusikan proses kerja secara pararel . Terdapat Driver Program, Cluster Manager dan Workers. Spark menjalankan aplikasi secara independen untuk memproses di sebuah kluster yang dikoordinasikan oleh driver program. Kemudian driver program yang terkoneksi dengan beberapa cluster manager akan mengalokasikan sumber daya ke seluruh aplikasi. Setelah terhubung, spark membutuhkan eksekutur yaitu workers pada node yang akan memproses jalannya komputasi (menggunakan bahasa phyton, java, scala) secara pararel dan menyimpan data.

Dalam Mlib Spark, beberapa algoritma mesin pembelajaran klasik tidak dimasukkan karena mereka tidak dirancang untuk platform paralel, tetapi berbeda MLlib berisi beberapa algoritma penelitian terbaru untuk cluster, seperti *distributed random forests, K-means*, dan *Alternating Least Square*. Dengan demikian MLlib paling cocok pada algoritma untuk menjalankan di dataset yang besar dengan beberapa iterasi. Berbeda dengan Weka yang menggunakan *single node* yang lebih cocok digunakan dataset berskala kecil.

2.6 Pengujian Mesin Rekomendasi

Sebuah sistem pengukuran dapat bernilai akurat dan tepat, atau akurat tetapi tidak tepat, atau tepat tetapi tidak akurat atau tidak tepat dan tidak akurat. Keakuratan metode estimasi kesalahan pengukuran diindikasikan dengan adanya error yang kecil. Metode estimasi yang mempunyai error lebih kecil dikatakan lebih akurat daripada metode estimasi yang mempunyai error lebih besar [37]. Presisi menunjukkan seberapa dekat perbedaan nilai pada saat dilakukan pengulangan pengukuran.



Gambar 2.10.Ilustrasi Perbedaan antara Akurasi dan Presisi

Dalam Gambar 2.9, pengukuran berulang diibaratkan dengan anak panah yang menembak target beberapa kali [38]. Akurasi menggambarkan kedekatan panah panah dengan pusat sasaran. Panah yang menancap lebih dekat dengan pusat sasaran dianggap lebih akurat. Semakin dekat sistem pengukuran terhadap nilai yang diterima, sistem dianggap lebih akurat.

Jika sejumlah besar anak panah ditembakkan, presisi adalah ukuran kedekatan dari masing-masing anak panah dalam kumpulan tersebut. Semakin menyempit kumpulan anak panah tersebut, sistem dianggap semakin presisi. Untuk

menentukan akurasi sebuah model ditentukan oleh uji validasi sedangkan untuk menentukan presisi hasil dari sebuah model ditentukan oleh uji presisi, dimana keduanya mempunyai metode tersendiri dalam pengujiannya.

2.6.1 Uji Validasi

Uji validasi bertujuan untuk menemukan parameter terbaik dari suatu model yang dilakukan dengan cara menguji besarnya error pada data testing. Hasil dari validasi menunjukkan tingkat akurasi sebuah model. Terdapat beberapa metode validasi yang dapat digunakan untuk menemukan model terbaik pada proses matriks faktorisasi ini :

a. *Mean Absolute Error* (MAE)

Nilai MAE merepresentasikan rata – rata kesalahan (*error*) absolut antara hasil peramalan dengan nilai sebenarnya.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{20}$$

dimana f_i adalah nilai hasil peramalan, y_i adalah nilai sebenarnya, dan n adalah jumlah data. Berdasarkan formula 10 di atas, MAE secara intuitif menghitung rata – rata error dengan memberikan bobot yang sama untuk seluruh data.

b. *Mean Squared Error* (MSE)

Nilai MSE dapat dianalogikan sebagai varian ditambah dengan kuadrat bias dari suatu model. Secara matematis MSE didefinisikan pada persamaan (21)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{21}$$

dimana f_i adalah nilai hasil peramalan, y_i adalah nilai sebenarnya, dan n adalah jumlah data. Berdasarkan formula 11 di atas, MSE memberikan bobot yang lebih besar jika dibandingkan dengan MAE, yakni nilai kuadratik dari error. Sebagai konsekuensinya, nilai error yang kecil akan semakin kecil dan nilai error yang besar akan semakin besar.

c. *Root Mean Squared Error* (RMSE)

RMSE menjadi alternatif yang lebih intuitif dibandingkan MSE karena memiliki skala pengukuran yang sama dengan data yang sedang dievaluasi. RMSE didapatkan rumus:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(fi - yi)^2} \quad (22)$$

dimana fi adalah nilai hasil peramalan, yi adalah nilai sebenarnya, dan n adalah jumlah data. Berdasarkan formula 12 di atas, RMSE memberikan bobot yang lebih besar jika dibandingkan dengan MSE, yakni nilai akar kuadratik dari error. Sebagai contoh, dua kali nilai RMSE artinya model memiliki error dua kali lebih besar dari sebelumnya. Sedangkan dua kali nilai MSE tidak berarti demikian. Jika MSE dapat dianalogikan sebagai varian, maka RMSE dapat dianalogikan sebagai standar deviasi.

2.6.2 Uji Presisi

Presisi merupakan salah satu pengujian dasar dan paling sering digunakan dalam penentuan efektifitas *information retrival system* maupun recommendation sistem [21] [39]. Presisi juga dapat digunakan untuk evaluasi kualitas hasil dari rekomendasi jika hasil prediksi diberikan filter baru dengan metode klasifikasi atau kemiripan dengan suatu konten [40] [39]. Untuk mengetahui kualitas hasil rekomendasi, dapat menggunakan rumus relevansi presisi yang membandingkan antara item yang relevan dengan total item yang dihasilkan atau yang direkomendasikan kepada user.

$$\text{Precission} = \frac{relevant\ item\ retrieved}{retrieved\ item} \quad (23)$$

Dimana relevant item retrieved adalah jumlah item relevan yang terpanggil, dan retrieved item adalah jumlah seluruh item relevan yang yang ditampilkan sebagai hasil rekomendasi. Suatu item dikatakan sangat presisi jika nilainya adalah 1 dan tidak presisi apabila nilainya menjauhi angka 1. *Threshold value* atau ambang batas menjadi penentu dari jumlah item yang ditemukan, semakin tinggi *threshold* maka jumlah item yang ditemukan sedikit. Semakin rendah threshold maka semakin banyak item yang ditemukan namun memiliki presisi yang rendah.

2.6.3 Uji Penerimaan User

User Acceptance Test (UAT) atau uji penerimaan user adalah pengujian hasil rekomendasi yang dilakukan oleh user bahwa seberapa besar hasl rekomendasi dapat diterima pengguna [21]. Pada pengujian ini bisa dilakukan dengan menyebarkan kuesioner yang berisikan pertanyaan mengenai kesesuaian antara hasil rekomendasi dengan teori yang telah dijabarkan dan menguji hasil rekomendasi apakah sesuai dengan selera pengguna. Untuk menguji hasil rekomendasi yaitu jumlah item yang sesuai dengan selera pengguna berbanding dengan jumlah dokumen yang ditampilkan dapat pula menggunakan rumus presisi (23) [21]. Hasil akhir dari kuesioner dapat memperkuat analisis mengenai kualitas dari mesin rekomendasi ini.

2.7 Penelitian yang berkaitan

Dalam tabel 2.1 ini adalah penelitian tentang mesin rekomendasi yang menggunakan metode collaborative filtering dengan dataset yang juga diambil dari dataset yang telah tersedia di lembaga riset atau web mesin rekomendasi yang telah ada.

Tabel 2.1 Penelitian yang berkaitan dengan metode collaborative filtering

No.	Judul dan Peneliti	Metode, Algoritma dan Datasets	Hasil yang didapatkan
1	A new collaborative filtering metric that improves the behavior of recommender systems.(J. Bobadilla , F. Serradilla, J. Bernal)	Collaborative fltering – Pearson Corellation dengan tambahan Jaccard measure dan traditional metric. Menggunakan dataset Movielends (1M ratings) dan Netflix (10M ratings)	MAE pada datasets Movie lends : pearson corellation 1.9 sementara matriks baru dengan penambagan jacccard meassure dan traditional matriks MAE turun menjadi 1.3. MAE pada datasets Movie lends : pearson corellation 0,75 sementara matriks baru dengan penambagan jacccard meassure dan traditional matriks MAE naik menjadi 0,95

Tabel 2.1 Penelitian yang berkaitan dengan metode collaborative filtering (Lanjutan)

No.	Judul dan Peneliti	Metode, Algoritma dan datasets	Hasil yang didapatkan
2	Item-Based Collaborative Filtering Recommendation Algorithms. (Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl)	Collaborative filtering (CF)- Matrix Faktorisasi dengan model-based method. Menggunakan datasets movielends 100K	Memperbaiki hasil metode memory-based collaboration filtering yang menggunakan algoritma KNN yaitu pada sparsity dan skalabilitas. Dengan matrik faktorisasi prediksi untuk seluruh item yang telah mendapat rating dibuat dan di rekomendasikan. Didapatkan MAE 0,75.
3	An Approach for Recommender System by Combining Collaborative Filtering with User Demographics and Items Genres (Saurabh Kumar Tiwari and Shailendra Kumar Shrivastava)	Collaboative filtering menggunakan KNN dan menggunakan cosine similiarity untuk mengaitkan dengan genre dan demografic (feature content). Menggunakan movielends datasets(100K ratings)	Menghasilkan RMSE 1,18 untuk collaborative filtering menggunakan KNN., namun cold start problem untuk pengguna baru dapat dipecahkan berdasarkan demografik pengguna, sedangkan item cold start dipecahkan dengan item cluster
4	An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering (Mustansar Ali Ghazanfar and Adam Pr¨ugel-Bennett)	Hybrid recommender system dengan menggunakan collaborative filtering (KNN) dan content based filtering (naive bayes classifier) dengan teknik switching. Menggunakan data set movielends (100K ratings) dan Filmtrust (28K ratings)	Hasil dari collaborative filering didapatkan MAE 1.421 dan hasil dari hybrid didapatkan ROC 0,657. Dapat memecahkan masalah cold start problem pada item baru walaupun MAE dinilai masih tinggi

Tabel 2.1 Penelitian yang berkaitan dengan metode collaborative filtering (Lanjutan)

No.	Judul dan Peneliti	Metode, Algoritma dan datasets	Hasil yang didapatkan
5	Large-scale Parallel Collaborative Filtering for the Netflix Prize. (Yunhong Zhou, Dennis Wilkinson, Robert Schreiber and Rong Pan	Memperkenalkan algoritma ALS-WR. Menggunakan netflix dataset (1,4M ratings)	Dapat mengatasi masalah sparsity, scalability dan overfitting dibandingkan dengan algoritma collaboratif filtering lainnya dengan pendekatan tetangga. Menghasilkan RMSE 0,935.
6	Machine Learning at Scale (Ondřej Fiedler)	Membandingkan performansi antara k-NN dan ALS pada mesin pembelajaran Mlib Spark untuk pembuatan rekomendasi film dengan metode collaborative filtering. Menggunakan Netflix dataset (100M ratings)	ALS menghasilkan RMSE yang lebih baik yaitu 0,45 dibandingkan dengan KNN yaitu 1,4 . Untuk training, ALS membutuhkan waktu lebih lama daripada KNN namun runtime untuk menghasilkakn rekomendasi ALS lebih cepat dibandingkan dengan KNN.

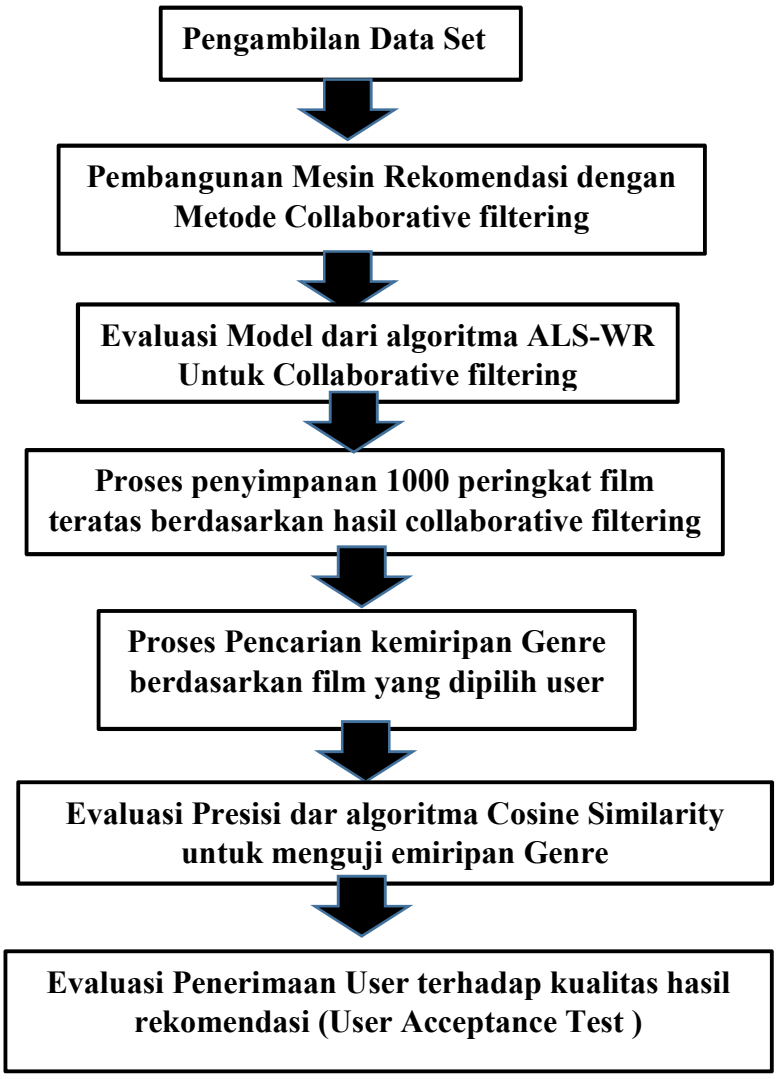
Halaman ini segaja dikosongkan

BAB 3
METODOLOGI PENELITIAN

Di dalam bab ini akan dijelaskan metode dan cara kerja yang akan digunakan dalam penelitian, sehingga dapat memberikan gambaran bagaimana membagun mesin rekomendasi dengan kemiripan genre berdasarkan metode collaborative filtering.

3.1 Alur Penelitian

Alur penelitian yang digunakan dalam penelitian ini terdiri dari 8 tahap yang digambarkan pada Gambar 3.1



Gambar 3.1. Diagram Alir Penelitian

3.2 Tahap Pengambilan Data Set

Data set diambil dari Movielends.org sebuah lembaga riset yang memfokuskan kajian tentang mesin rekomendasi. Dataset yang dipilih adalah Movielends yaitu riwayat transaksi rental video online. Peneliti mengambil 3 dataset berukuran 100K, 1M dan 10M dimana masing-masing dataset memiliki rincian jumlah film, user dan item yang ditunjukkan pada tabel 3.1

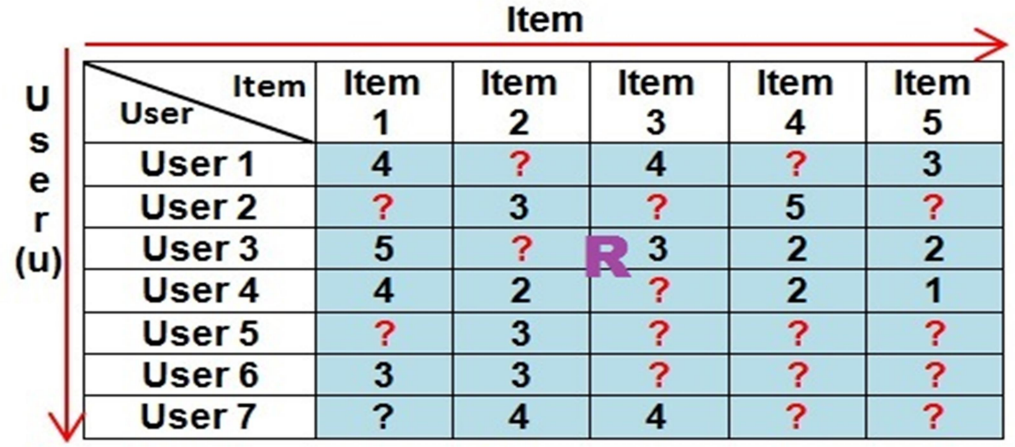
Tabel 3.1 Jumlah film, rating dan user untuk 3 dataset

Size Dataset	Jumlah film	Jumlah Rating	Jumlah User	Sparsitas
100K	10.681 movies	100.004 ratings	671 user	98.6%
1M	3952 movies	1.000.209 rating	6040 user	95.8%
10M	10.681 movies	10.000.054 rating	71.567 user	98.6%

Pada Tabel 3.1 kolom 1 menjelaskan nama data set, 100K berarti dataset tersebut berisi kurang lebih 100 Ribu rating, sementara 1M adalah 1 milion yaitu dataset tersebut meliki rating kurang lebih 1 Juta rating, sementara 10M adalah 10 Milion yang berarti data set tersebut memiliki kurang lebih 10 Juta rating. Kolom kedua yaitu jumlah film menunjukkan jumlah film yang dimiliki tiap dataset. Kolom 3 menunjukkan rincian jumlah rating yang dimiliki tiap dataset. Kolom ke 4 menunjukkan jumlah user dan kolom kelima menunjukkan tingkat sparsitas dataset. Tingkat sparsitas artinya bahwa tidak semua film mendapatkan rating oleh seluruh user seperti yang ditunjukkan pada Gambar 3.2

Tingkat sparsitas pada masing-masing dataset didapatkan dari jumlah rating jika seluruh item terisi penuh dikurangi jumlah rating yang sebenarnya dikalikan 100%, dapat dicontohkan perhitungannya pada rumus (24)

$$\text{Sparsitas pada Dataset} = \frac{(\text{Jumlah user x Jumlah Movie}) - \text{Jumlah Rating}}{\text{Jumlah user x Jumlah Movie}} \times 100\% \qquad (24)$$



Gambar 3.2 Sparsitas atau nilai yang kosong dari matriks user-item yang ditunjukkan dengan tanda tanya berwarna merah (?)

Pada dataset 100K sparsitas yang ditunjukkan adalah 98,6% berarti sekitar 98,6% kolom matriks user-item belum memiliki nilai rating, pada dataset 1M sparsitas yang ditunjukkan adalah 95,8% berarti sekitar 95,8% kolom matriks user-item belum memiliki nilai rating, pada dataset 100K sparsitas yang ditunjukkan adalah 98,6% berarti sekitar 98,6% kolom matriks user-item belum memiliki nilai rating inilah yang coba dipecahkan dengan metode collaborative filtering dengan algoritma ALS-WR.

Pada dataset ini memiliki format '.dat' artinya data dipisahkan dengan dua titik (::) dengan struktur sebagai berikut :

a. movie.dat dengan struktur :MovieID::Title::Genres
   contoh pada Gambar 3.2
b. rating.dat dengan struktur : UserID::MovieID::Tag::Timestamp.
   contoh pada Gambar 3.3
c. User.dat dengan struktur : UserID::Gender::Age::Occupation::Zip-code
   contoh pada Gambar 3.4

1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance
4::Waiting to Exhale (1995)::Comedy|Drama
5::Father of the Bride Part II (1995)::Comedy
6::Heat (1995)::Action|Crime|Thriller
7::Sabrina (1995)::Comedy|Romance
8::Tom and Huck (1995)::Adventure|Children's
9::Sudden Death (1995)::Action
10::GoldenEye (1995)::Action|Adventure|Thriller
11::American President, The (1995)::Comedy|Drama|Romance
12::Dracula: Dead and Loving It (1995)::Comedy|Horror
13::Balto (1995)::Animation|Children's
14::Nixon (1995)::Drama
15::Cutthroat Island (1995)::Action|Adventure|Romance
16::Casino (1995)::Drama|Thriller
17::Sense and Sensibility (1995)::Drama|Romance
18::Four Rooms (1995)::Thriller
19::Ace Ventura: When Nature Calls (1995)::Comedy
20::Money Train (1995)::Action
21::Get Shorty (1995)::Action|Comedy|Drama

Gambar 3.3 Contoh dataset Movie.dat

1::1193::5::978300760
1::661::3::978302109
1::914::3::978301968
1::3408::4::978300275
1::2355::5::978824291
1::1197::3::978302268
1::1287::5::978302039
1::2804::5::978300719
1::594::4::978302268
1::919::4::978301368
1::595::5::978824268
1::938::4::978301752
1::2398::4::978302281
1::2918::4::978302124
1::1035::5::978301753
1::2791::4::978302188
1::2687::3::978824268

Gambar 3.4 Contoh dataset Movie.dat

1::F::1::10::48067
2::M::56::16::70072
3::M::25::15::55117
4::M::45::7::02460
5::M::25::20::55455
6::F::50::9::55117
7::M::35::1::06810
8::M::25::12::11413
9::M::25::17::61614
10::F::35::1::95370
11::F::25::1::04093
12::M::25::12::32793
13::M::45::1::93304
14::M::35::0::60126
15::M::25::7::22903
16::F::35::0::20670
17::M::50::1::95350

Gambar 3.5 Contoh dataset Movie.dat

Pada Gambar 3.3 yaitu dataset movie.dat terlihat Genre dari film-film tersebut. Genre adalah klasifikasi dari kandungan kontent. Genre tersebut dibuat berdasarkan review para ahli perfilman. Pada data set ini Genre telah tersedia. 1 film memiliki satu atau beberapa Genre. Genre-genre tersebut pada dataset ini dipisahkan dengan pipa pemisah seperti yang ditunjukkan pada Gambar 3.3. Daftar lengkap genre dapat dilihat pada file Readme.html adalah sebagai berikut :

- Action
- Adventure
- Animation
- Children's
- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western

Terdapat total 17 genre dari film-film pada dataset movielends ini.

Dalam penelitian ini, peneliti hanya menggunakan 2 dataset yaitu movie.dat dan rating.dat. Untuk *collaborative filtering* dari movie.dat yang digunakan adalah *movie id* dan *title* namun hasil yang di print genre tetap dimasukkan karena diperlukan untuk proses selanjutnya yaitu dalam pembuatan kemiripan genre, dan dari rating dat id user, id movie dan rating dibutuhkan. Untuk pembuatan rekomendasi kedua tidak menggunakan raw data (data dari data set) tetapi data yang

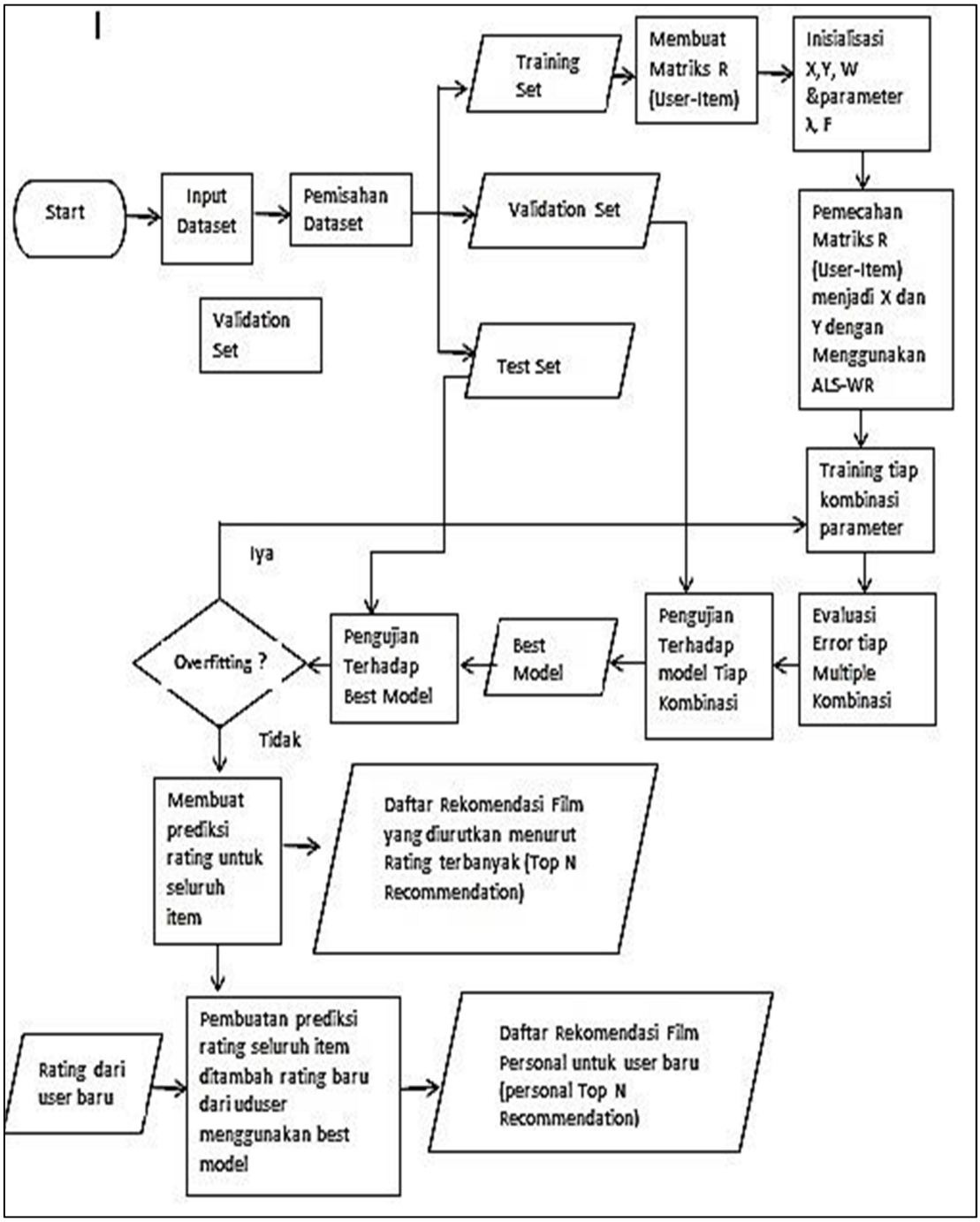
telah diolah dan diranking pada *collaborative filtering*. Hasil dari *collaborative filtering* diambil hanya 1000 data yang terdiri atas id movie, title movie dan genre, kemudian diolah menggunakan cosine similiarity sesuai dengan pilihan movie yang dipilih user.

3.3 Pembangunan Mesin Rekomendasi dengan Metode Collaborative filtering

Pada tahap ini model mesin rekomendasi dan alur kerjanya diperkenalkan dan dijelaskan untuk menjawab permasalahan dan tujuan penelitian. *Collaborative filtering* seperti yang dijelaskan diatas bekerja berdasarkan interaksi antara pengguna dan item dimana interaksi keduanya kemudian menghasilkan sebuah profil yang dapat diolah untuk menghasilkan personal rekomendasi untuk user tresebut. Terdapat dua buah dalam pengambilan data, yaitu pendekatan implicit dan explicit. Pendekatan implicit, artinya, sistem menyimpan dan mempelajari perilaku pengguna terhadap item, contohnya item apa yang pernah dibeli pengguna, berapa kali penggula melihat barang tersebut, dsb. Sementara pendekatan explicit, yaitu dengan menanyakan kepada pengguna secara langsung deskripsi item yang bagaimana yang ia sukai/minati contoh keluarannya berupa rating atau kuesioner [23]. Model mesin rekomendasi pada penelitian ini dibangun berdasarkan data eksplisit dan feedback implisit yang diberikan user atas item yang telah dibelinya sebagaimana dijelaskan pada diagram alir pada Gambar 3.6. Dalam penelitian ini yang menggunakan metode *collaborative filtering* mengambil data eksplisit dari rating. Rating yang dimaksud disini adalah penilaian yang diberikan user terhadap item yang telah dibeli atau dilihat, lalu kemudian mesin pembelajaran dengan algoritma ALS-WR akan mengolah rating tersebut untuk mencari hubungan antar item berdasarkan tabel rating untuk membentuk sebuah rekomendasi terhadap suatu item kepada user.

Pada Gambar 3.6 menggambarkan diagram alir pada mesin rekomendasi dalam menghasilkan rekomendasi menggunakan algoritma ALS-WR. Pada proses awal collaborative filtering, yaitu input data ke Spark, kemudian data akan masuk ke *Resilient Distributed Dataset* (RDD). RDD berfungsi sebagai media pennyimpanan dan pembelajaran pada mesin learning spark. Pada RDD, manager konteks diciptakan, hal tersebut berguna agar proses kerja dapat dilaksanakan secara pararel sehingga waktu yang dibutuhkan relatif cepat karena dalam penciptaan model nantinya iterasi akan dilakukan berulang-ulang untuk mencapai titik konvergen.

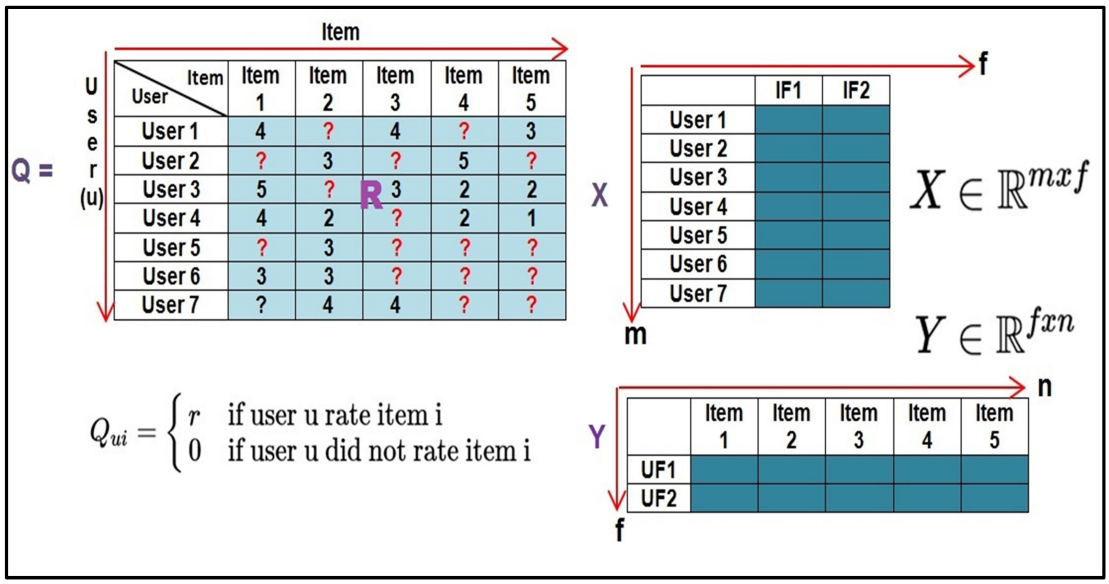
Setelah manajer konteks diciptakan, dataset dibagi 3 (spare set) untuk 60% data untuk train, 20% dataset untuk validasi dan dan 20% dataset untuk test. Pembagian dataset tersebut secara random.



Gambar 3.6 Digram Alir metode *collaborative filtering*

Kemudian dilakukan proses training dilakukan yaitu pertama dengan memetakan daset kedalam matriks user-item seperti pada gambar 3.2. Beberapa paramater

dimasukkan yaitu beberapa paramater *rank* (w) dan *lambda* (λ). Dalam algoritma ALS-WR, matriks user item dipecah menjadi dua yaitu matriks X yang berukuran *mxf* dan matriks Y yang berukuran *fxn* seperti pada Gambar 3.7.



Gambar 3.7 Pemecahan matriks user-item menjadi matriks X dan Y

Dengan menggunakan persamaan (14) dan (15) akan membuat multiple kombinasi dengan beberapa parameter tersebut rank dan lambda. Hasilnya kemudian di validasi dengan data validasi untuk menghasilkan RMSE terkecil. Best model yang didapatkan kemudian di teskan kembali ke data set untuk membuktikan tidak ada overfitting dari proses tersebut. Best model kemudian digunakan untuk menghasilkan prediksi rekomendasi. Prediksi ini menghasilkan prediksi rating untuk setiap item untuk mengisi sparsitas pada dataset dengan demikian semua item mendapatkan rating dari semua user. Kemudian setiap item total jumlah ratingnya dan diurutkan/diranking dari item yang memiliki jumlah rating terbesar hingga terkecil. Daftar rekomendasi ini kemudian menghasilkan daftar rekomendasi film berdasarkan rating terbanyak atau yang disebut TopN. Setelah TopN dihasilkan selanjutnya adalah pembuatan rekomendasi personal yaitu rekomendasi yang disusun berdasar personal user dalam penelitian ini adalah user baru yang profilnya belum terdapat di datasets. Mesin pembelajaran kemudian akan menambahkan rating yang diberikan oleh user baru tersebut untuk kemudian dibuatkan prediksi kembali dengan

menggunakan best model hasil training tadi lalu hasilnya diurutkan kembali menjadi TopN yang sifatnya personal (personal TopN Recommendation).

3.4 Uji Validitas *Collaborative Filtering*

Seperti penjelasan diatas bahwa ALS-WR yang digunakan untuk collaborative filtering adalah algoritma matrik faktorisasi dengan beberapa kali iterasi, sehingga mendapatkan beberapa nilai alternatif, dan model terbaik didapatkan pada saat titik konvergen telah tercapai dan error menjadi minimal. Sebelum mencapai model terbaik, terlebih dahulu training pada mesin pembelajaran Mlib Apache Spark. Dalam proses training ini, data dibagi 3 secara random, terpisah dan tidak tumpang tindih. Proses *training* ini adalah train, validation dan test. Pada proses awal, mlib apache spark akan melatih multiple model berdasarkan training set, memilih model terbaik pada *validation set* dan pada akhirnya mengevaluasi model terbaik pada test set. Outputnya adalah Root Mean Squared Error yaitu kuadrat rata-rata error terkecil yang dihasilkan dari perhitungan model yang dihasilkan. RMSE menggunakan rumus :

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(fi - yi)^2} \qquad (12)$$

dimana fi adalah prediksi rating dan yi adalah rating yang sebenarnya, dan n adalah jumlah data. Proses RMSE ini dilakukan dua kali yaitu pada saat validasi dan pada saat test. Kemudian kedua nilai ini dapat dibandingkan untuk membuktikan ada atau tidak overfitting. Jika overfitting terjadi maka training dan pencarian model terus dilakukan sampai RMSE tidak overfitting.

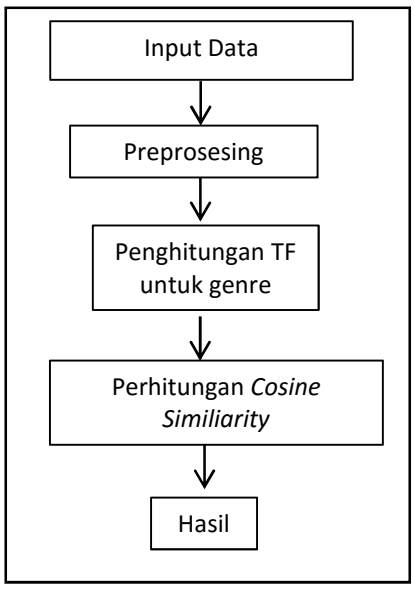
3.5 Proses penyimpanan 1000 peringkat film teratas hasil collaborative filtering

Untuk meningkatkan kualitas dari collaborative filtering, maka korelasi dari content tetap dibutuhkan contohnya film memiliki informasi bergenre yang diberikan oleh para ahli film dan sutradara. Terkadang, informasi kategori ini lebih disukai daripada penilaian pengguna yang memiliki selera yang sama [32]. Berdasarkan hal tersebut, berdasarkan hasil collaborative filtering kemudian diambil 1000 daftar film dan genre dari 1000 peringkat teratas rekomendasi personal untuk dimasukkan kedalam proses selanjutnya yaitu mencari kemiripan tiap genre dari film yang dipilih

user pada tahap ini. Proses penyimpanannya yaitu dengan membuat database di SQL yang beisi daftar movie dan genre. Database ini sifatnya temporary, artinya setiap ada user baru yang menambahkan rating, maka mesin learning akan kembali mengenerate personal rekomendasi dan menyimpannya kembali di SQL.

3.6 Kemiripan Genre

Setelah hasil rekomendasi pertama berdasarkan collaborative filtering disimpan, dsi halaman interface, user yang telah mendapatkan hasil rekomendasi kemudian dapat memilih/mengklik item yang disukainya untuk melihat detail item tersebut, dari proses ini menghasilkan data implisit yang menjadi dasar pembuatan rekomendasi tahap kedua. Alur kerja proses mencari kemiripan film berdasarkan genre dijelaskan pada Gambar 3.8.



Gambar 3.8. Alur Kerja proses kemiripan genre menggunakan cosine similarity

a. Input Data

Di halaman interface, user yang telah mendapatkan hasil rekomendasi kemudian dapat memilih/mengklik item yang disukainya untuk melihat detail item tersebut, dari proses ini menghasilkan data implisit yang menjadi dasar pembuatan rekomendasi tahap kedua.

Tabel 3.2 Daftar film berikut genrenya hasil dari collaborative filtering.

No. Film	Daftar Film	Genre
D.1	Star Trek	war, action, science-fi, adventure
D.2	Avatar	animation, romance, war, adventure
D.3	Braveheart	war, action, romance
D.4	Star Wars	war, action, science-fi, adventure
D.5	**Jurastic Park**	**adventure, action**
D.6	Alice and Wonderland	amination, drama. Adventure

Pada Tabel 3.2 adalah contoh daftar film berikut genrenya hasil dari collaborative filtering daftar film berikut genrenya hasil dari collaborative filtering. Arsir kuning adalah film yang kemudian dipilih oleh user yaitu film Jurastic Park dengan genre Adventure dan Action. Kedua Genre itulah yang kemudian menjadi dasar perhitungan cosine similarity untuk mencari rekomendasi film dengan genre yang mirip.

b. Preprosesing

Tahap berikutnya adalah preprosesing. Dalam proses ini seluruh data akan diubah semua tulisan menjadi huruf kecil yang disebut dengan istilah *case folding*. Setelah proses case folding yaitu *tokenizing*, pada proses ini dihapus semua karakter seperti angka, tanda baca, symbol dan memisahkan kalimat menjadi kata atau disebut juga dengan *parsing*.

b. Pembobotan TF-IDF untuk Genre

Setelah melakukan tahap *pre-processing* adalah proses pembobotan dengan menghitung frekuensi kemunculan genre dalam satu film dan seluruh seluruh daftar film yang dijadikan data uji serta data itu sendiri yang akan dklasifikasikan. Film diibaratkan sebagai Dokumen, satu film adalah 1 dokumen, dengan demikian pre-processing adalah proses pembobotan dengan menghitung frekuensi kemunculan genre dalam satu dokumen dan seluruh seluruh daftar dokumen yang dijadikan data uji serta data itu sendiri yang akan dklasifikasikan. Rumus dari proses ini adalah "W$dt = tf$*idf", dimana "idf=Log(n/df)", n adalah jumlah data uji ditambah data Q, tf adalah kata dasar yang muncul dalam satu data, dan df adalah jumlah kata dasar yang muncul dalam satu . Pada contoh ini Query adalah genre dari film jurastic park yaitu Adventure dan Action.

Contoh

Diketahui : Query adalah genre dari film jurastic park yaitu Adventure dan Action.

Dokumen 1 (D1) Star Trek. Genre : war, action, science-fi, adventure

Tf untuk term Action =1

Df yaitu kemunculan kata action untuk seluruh dokumen adalah 5 dari n yaitu total seluruh term dari seluruh dokumen

Nilai rata frekuensi untuk kata action $(\frac{n}{df})$ dengan demikian adalah: $\frac{7}{5} = 1.40$Terkadang suatu term muncul di hampir sebagian besar dokumen mengakibatkan proses pencarian term unik terganggu. Idf berfungsi mengurangi bobot suatu term jika kemunculannya banyak tersebar di seluruh koleksi dokumen kita. Rumusnya adalah dengan inverse document frequency. Document frequency adalah seberapa banyak suatu term muncul di seluruh document yang diselidiki. Rumus Idf adalah $\text{Log}\frac{n}{df}$ dengan demikian cara menghitungnya adalah $\text{Log}\frac{7}{5} = 0.146$. Log atau logarritmik diperlukan untuk mengurangi besarnya bilangan, dimana logaritmik suatu bilangan akan mengurangi digit jumlah, contoh 1000 dengan log (1000) hanya menghasilkan angka tiga.

Pembobotan Tf-Idf (Wdt) adalah perkalian Tf x Idf yang ditunjukkan pada Tabel 3.4 Untuk Dokumen 1 nilai Wdt nya adalah 0.14 x 1 = 0.146. Dengan demikian bobot tTf-Idf kata action pada dokumen satu bernilai 1.4. Hasil lengkap seluruh dokumen dapat dilihat pada Tabel 3.3 dan Tabel 3.4

Tabel 3.3 Contoh Hasil perhitungan Tf dan Idf

No.	Term	Q	Tf									idf
			D1	D2	D3	D4	D5	D6	df	n	n/df	Log(n/df)
1	war	0	1	1	1	1	0	0	4	7	1.75	0.24304
2	action	1	1	0	1	1	1	0	5	7	1.40	0.14613
3	science-fi	0	1	0	0	1	0	0	2	7	3.50	0.54407
4	adventure	1	1	1	0	1	1	1	5	7	1.40	0.14613
5	animation	0	0	1	0	0	0	1	2	7	3.50	0.54407
6	romance	0	0	1	1	0	0	0	2	7	3.50	0.54407
7	drama	0	0	0	0	0	0	1	2	7	3.50	0.54407

Tabel 3.4 Contoh Hasil pembobotan Tf-Idf

No.	Term	Q	***Wdt = tf.idf***						
			Q	D1	D2	D3	D4	D5	D6
1	war		0	0.24	0.24	0.24	0.24	0	0
2	action		1.4	0.15	0	0.15	0.15	0.15	0
3	science-fi		0	0.54	0	0	0.54	0	0
4	adventure		1.4	0.15	0.15	0	0.15	0.15	0.15
5	animation		0	0	0.54	0	0	0	0.54
6	romance		0	0	0.54	0.54	0	0	0
7	drama		0	0	0	0	0	0	0.54

c. Perhitungan Cosine Similarity

Cosine Similarity adalah menghitung tingkat kemiripan vector data Q dengan data uji. Kemiripan antar dokumen *Cosine Similarity* menggunakan rumus (16) yang telah dipaparkan pada Bab.2. Langkah dalam menghitung rumus *Cosine Similarity* adalah sebagai berikut :

hitung hasil perkalian scalar antara Q dan data uji. Hasilnya perkalian dari setiap film dengan Q dijumlahkan (sesuai pembilang pada rumus cosine similarity)

hitung panjang setiap dokumen, termasuk Q. Caranya kuadratkan bobot setiap *term* dalam setiap dokumen, jumlahkan nilai kuadrat dan terakhir akarkan.

Tabel 3.5 Contoh hasil perhitungan cosine Similiarity

No.	Term	**W(Q) * W(Di)**						**W(Q) * W(Di)**						
		D1	D2	D3	D4	D5	D6	Q	D1	D2	D3	D4	D5	D6
1	war	0	0	0	0	0	0	0.00	0.06	0.06	0.06	0.06	0	0
2	action	0.2	0	0.2	0.2	0.2	0	1.96	0.02	0	0.02	0.02	0.02	0
3	science-fi	0	0	0	0	0	0	0.00	0.3	0	0	0.3	0	0
4	adventure	0.2	0.2	0	0.2	0.2	0.2	1.96	0.02	0.02	0	0.02	0.02	0.02
5	animation	0	0	0	0	0	0	0.00	0	0.3	0	0	0	0.3
6	romance	0	0	0	0	0	0	0.00	0	0.3	0.3	0	0	0
7	drama	0	0	0	0	0	0	0.00	0	0	0	0	0	0.3
		0.4	0.2	0.2	0.41	0.4	0.2	3.92	0.4	0.67	0.38	0.4	0.04	0.61
								1.98	0.63	0.82	0.61	0.63	0.21	0.78

Langkah selanjutnya yaitu menerapkan rumus *Cosine Similarity*. Hitung kemiripan Q dengan data uji (D1, D2, D3, D4, D5, D6).

Cos (Q, D1) = 0.41/(1.98*0.63) = 0.327

Cos (Q, D2) = 0.20/(1.98*0.82) = 0.126

Cos (Q, D3) = 0.20/(1.98*0.61) = 0.168

Cos (Q, D4) = 0.40/(1.98*0.63) = 0.327

Cos (Q, D5) = 0.40/(1.98*0.21) = 1

Cos (Q, D6) = 1.20/(1.98*0.78) = 0.131

Maka peringkat dokumen dari Dokumen 1-Dokumen 6 yang paling dekat dengan quary ditunjukkan pada Tabel 3.6

Tabel 3.6 Contoh film yang telah diurutkan menurut kemiripan genrenya

No. Peringkat	Daftar Film	Genre	Nilai Cosine
1	Jurastic Park	adventure, action	1
2	Star Wars	war, action, science-fi, adventure	0.327
3	Star Trek	war, action, science-fi, adventure	0.327
4	Braveheart	war, action, romance	0.168
5	Alice and Wonderland	amination, drama. Adventure	0.131
6	Avatar	animation, romance, war, adventure	0.126

3.7 Uji Presisi Cosine Similarity

Pengujian dilakukan untuk penilaian akhir rekomendasi dengan melibatkan user sebagai agen. User memilih salah satu item, kemudian mesin bekerja dengan algoritma *cosine similiarity* untuk menemukan item film yang memiliki genre yang mirip. Hasil dari rekomendasi akhir ini kemudian yang menjadi alat uji. Untuk mengetahui kualitas hasil rekomendasi, dapat menggunakan rumus relevansi presisi yang membandingkan antara item yang relevan dengan total item yang dihasilkan atau yang direkomendasikan kepada user.

$$\text{Presisi} = \frac{relevant\ item\ retrieved}{retrieved\ item} \qquad (23)$$

Dimana relevant item retrieved adalah jumlah item relevan yang terpanggil, dan retrieved item adalah jumlah seluruh item relevan yang yang ditampilkan sebagai hasil rekomendasi. Contoh mengunakan Tabel 3.6, diketahui jumlah film yang bernilai 1 adalah 1 film, jumlah yang ditampilkan adalah 6, maka nilai presisi didapatkan adalah $\frac{1}{6} = 0{,}166$

Jika thrshold dibuat bervariasi maka nilai presisi akan meningkat dan dokumen yang ditemukan akan semakin banyak.

3.8 Uji Penerimaan User

Dalam penelitian ini, Spark dan Mlib Spark 2.1.0 di instal di ubuntu 16.10 pada komputer dengan prosessor core i3 yang memiliki RAM 8GB. Algoritma ditulis pada phycharm dengan bahasa phyton sementara untuk interface dan perhitungan cosine similiarity menggunakan phpstorm dengan bahasa php. Graphical User Interface (GUI) juga menggunakan php karena pemrograman PHP versi 5 karena PHP dapat membangun sebuah sistem berbasis web yang ringan dan handal dan banyak digunakan di banyak website e-commerce sehingga model ini mudah diimplementasikan dan diaplikasikan.

Setelah engine rekomender dibuat untuk menjalankan ALS-WR. Spark dan mlib spark terlebih dahulu dijalankan pertamakali di terminal, setelah Top N recommendation berhasil didapatkan kemudian disimpan di SQL, karena Spark sendiri memiliki API yang dapat terhubung dengan SQL.User baru dapat memasukkan rating melalui GUI. Dengan menggunakan php shell, php dapat terhubung dengan Mlib Spark, jadi setelah user selesai merating dan mengklik tombol “run” secara otomatis spark berjalan untuk memproses rekomendasi I dengan algoritma ALS-WR. Setelah daftar rekomendasi I keluar, user dapat mengklik salah satu judul film kemudian proses cosine similiarity berjalan kemudian daftar rekomendasi II dapat keluar.

Uji Penerimaan User atau *User Acceptance Test* (UAT) dimaksudkan untuk menguji tingkat penerimaan user terhadap hasil rekomendasi. Untuk collaborative filtering pada data set 1M dipasang paramater best model dengan rank 10, lambda 0,8 dan untuk cosine similarity diberikan batas threshold 50% atau 0,5. Kemudian hasil rekomendasi dari parameter yang telah di set ini diberikan kepada 20 orang user. User mencoba memberikan rating untuk item film yang pernah dia sukai atau pernah ditonton melalui interface mesin rekomendasi , kemudian diberikan kuesioner yang terdiri dari 5 pertanyaan yang dapat dilihat pada Tabel 3.6.

Untuk menguji jumlah item film yang relevan menurut selera user dibandingkan dengan seluruh dokumen yang ditampilkan dapat pula menggunakan

rumus presisi Dari 5 pertanyaan tersebut dapat dianalisis hasilnya sebagai penerimaan user terhadap kualitas rekomendasi yang dihasilkan mesin rekomendasi ini

Tabel 3.6 Daftar Pertanyaan Uji Penerimaan User untuk Mesin Rekomendasi

No.	Daftar Pertanyaan	Jawaban	
1	Dari daftar rekomendasi pertama, apakah memberikan hasil yang tidak terduga (mengejutkan) yang anda tidak menyangka itu ada ternyata film tersebut tersedia ?	Ya	Tidak
2.	Dari 10 daftar Rekomendasi I, berapa jumlah rekomendasi yang relevan menurut selera anda ?		
3.	Klik salah satu dari 10 daftar yang dihasilkan dari rekomendasi metode I, kemudian lihat 10 daftar rekomendasi selanjutnya. Apakah memberikan hasil yang tidak terduga ?	Ya	Tidak
4.	Dari 10 daftar rekomendasi yang tersedia pada lembar rekomendasi II, berapa jumlah rekomendasi yang relevan menurut selera Anda ?		
5.	Rekomendasi yang manakah yang lebih baik menurut selera Anda diantara kedua hasil rekomendasi tersebut ?	Rekomendasi I	Rekomendasi II

3.7 Tahap Penarikan Kesimpulan

Pada tahap ini diketahui tingkat keakurasian prediksi yang ditunjukkan dengan error untuk 3 dataset yaitu 100K, 1M dan 10ML dan juga diketahui nilai presisi untuk hasil akhir rekomendasi dengan beberapa nilai *threshold*. Hasilnya akan diketahui apakah model ini dapat digunakan untuk user baru untuk memperoleh hasil rekomendasinya dan juga dapat diketahui apakah model ini dapat mengatasi skalabilitas pada data dengan jumlah yang terus meningkat. Selain itu menggunakan UAT yang diujikan kepada user dapat dilihat pula tingkat penerimaan user terhadap hasil rekomendasi yang diberikan oleh mesin rekomendasi ini

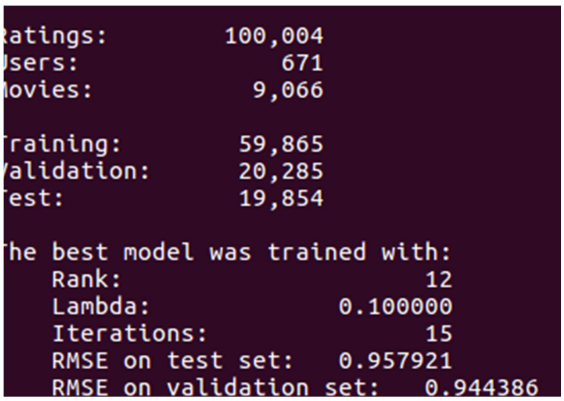
BAB 4
HASIL DAN PEMBAHASAN

4.1 Pengujian *Collaborative Filtering*

Pada subbab ini akan dijabarkan hasil dari pengujian 3 dataset menggunakan metode collaborative filtering dengan algoritma ALS-WR serta performansi ALS-WR dalam proses collaborative filtering untuk menghasilkan rekomendasi bagi pengguna.

4.1.1 Pengujian *Collaborative Filtering* pada dataset 100K

Pengujian terhadap collaborative filtering dilakukan pada saat awal yaitu pada saat training. Pada dataset 100K berkisar 10, 11 dan 12 dan lambda yaitu 0.7,0.8, 0.9, 0.1 dan iterasi hingga 25. Kemudian spark secara otomatis membagi tiap dataset menjadi 3 secara random, terpisah dan tidak tumpang tindih. 3 data tersebut digunakan untuk train, validation dan test. Dengan rumus yang dijabarkan pada persamaan pada (14) dan (15) mesin pembelajaran spark berusaha mencari model terbaik dari kombinasi parameter yang telah ditetapkan. Hasil dari proses training kemudian dapat dilihat pada Gambar 4.1.



Gambar 4.1. *Hasil training dari Dataset 100K*

Dari gambar 4.1 terlihat bahwa data berhasil dibagi menjadi tiga bagian yaitu 59.275 untuk training, 20.285 untuk validasi dan 19.854 rating yang digunakan untuk test. Best model yang didapatkan ada pada rank 12, labda 0.1 dengan iterasi 15 mampu menghasilkan RMSE 0.944 pada validation dan 0.957 pada test set. Dapat

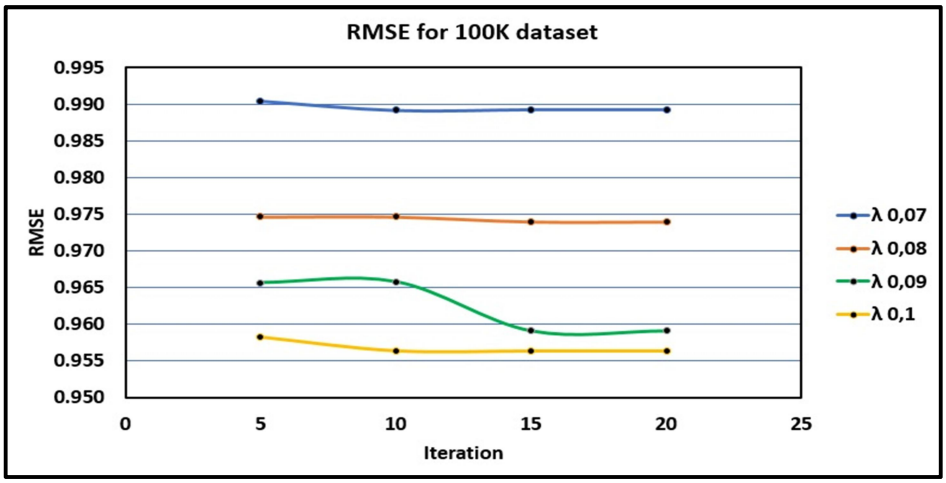
dilihat bahwa hasil test dan validasi tidak overfitting karena RMSE antara validasi dan test tidak terlampau jauh.

Pengujian selanjutnya adalah peneliti ingin mengetahui sejauh mana terjadinya perubahan error terhadap parameter yang berbeda di setiap iterasi hingga mencapai titik konvergen. Peneliti kemudian menguji satu persatu paramater yaitu lambda dan rankpada dataset 100K. Hasil yang didapatkan pada dataset 100K dapat dilihat pada Tabel 4.1 dan Gambar 4.1.

Tabel 4.1 Hasil generate data 100K

Lambda	Iterasi	RMSE	Lambda	Iterasi	RMSE
λ 0,07	5	0.990	λ 0,09	5	0.966
	10	0.989		10	0.966
	15	0.989		15	0.959
	20	0.989		20	0.959
λ 0,08	5	0.975	λ 0,1	5	0.958
	10	0.975		10	0.956
	15	0.974		15	0.956
	20	0.974		20	0.956

Dari Tabel 4.1 terlihat bahwa paramater yang berbeda akan menghasilkan RMSE yang berbeda. Terlihat bahwa nilai lambda mempengaruhi hasil RMSE, semakin banyak iterasi data juga semakin terlatih yang menyebabkan RMSE menjadi semakin kecil namun ketika telah mencapai titik konvergen maka walaupun terus melakukan iterasi namun hasilnya tidak ada perubahan secara signifikan atau bahkan tidak berubah sama sekali. Terlihat bahwa best model berhasil didapatkan pada lambda 0.1 dan iterasi ke 15. Titik konvergen akan terlihat lebih jelas pada grafik yang ditunjukkan pada Gambar 4.1.



Gambar 4.2 Grafik RMSE untuk 100K dataset

Pada gambar 4.1 terlihat bahwa titik konvergen terjadi pada iterasi ke 15. Setelah iterasi ke 15, perubahan nilai RMSE terlihat tidak signifikan atau tidak berubah samasekali. Sementara hasil rekomendasi yang dihasilkan dapat dilihat pada tabel 4.2. dan Tabel 4.3.

Tabel 4.2 Hasil TopN dari dataset 100K

Dataset	Ranking	Bobot	id movie	Title
100K	1	341	356	Forrest Gump
	2	324	296	Pulp Fiction
	3	311	318	Shawshank Redemption, The
	4	304	593	Silence of the Lambs, The
	5	291	260	Star Wars: Episode IV – A New Hope (a.k.a. Star Wars)
	6	274	480	Jurassic Park
	7	259	2571	Matrix, The
	8	247	1	Toy Story (1995)
	9	244	527	Schindlers List (1993)
	10	237	589	Terminator 2: Judgment Day

Tabel 4.3 Hasil Personal Rekomendasi dari dataset 100K

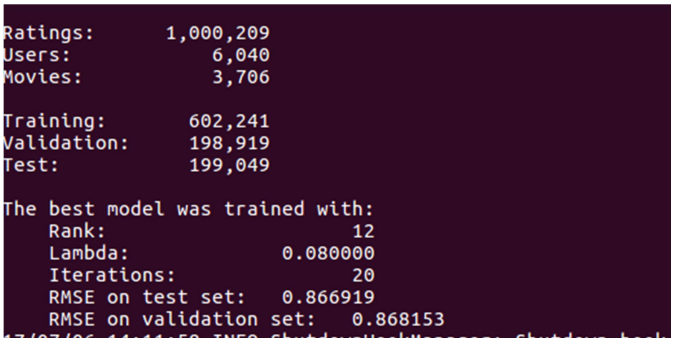
ALS-WR				
Input			Output	Validation
Data Set	**Title Movie**	**Rating**	**Title Movie**	**RMSE**
100K	Titanic	4	Fearless	
	Jurastic Park	2	Mr. Wonderful	**Best Model**
	The Matrix	2	Firm, The	λ =0.1,
	Toy Story	2	Color of Night	Rank =10
	Home Alone	3	Endless Summer 2	15 Iteration
	City of Angles	4	With Honors	**RMSE =**
	Breaveheart	3	It Takes Two	**0.957**
	Starwars	2	Bio-Dome	
	Something to talk about	3	Persuasion	
	Miracle on 34th street	3	Low Down Dirty Shame, A	

Tabel 4.2 adalah daftar rekomendasi yang diurutkan berdasarkan jumlah rating dari tiap film. Rating pada Top N ini adalah rating yang sebenarnya dan rating prediksi yang mengisi kolom matriks user-item. Seluruh rating tersebut di jumlahnya dan diperingkat menurut besanya jumlah rating. Berbeda dengan rekomendasi personal yang ditunjukkan pada Tabel 4.3 yaitu rekomendasi untuk user yang telah memasukkan rating film sebelumnya. Data dilihat kolom input dan kolom output. Pada pengujian mesin, peneliti mencoba memasukkan rating beberapa film yang disukai, lalu dengan best model, ALS-WR membuat rekomendasi personal dan menghasilkan rekomendasi film seperti yang bisa dilihat pada kolom output.

4.1.2 Pengujian Collaborative Filtering pada dataset 1M

Pengujian terhadap collaborative filtering dilakukan pada saat awal yaitu pada saat training. Pada dataset 1M diberikan parameter yaitu 10, 11 dan 12 dan lambda yaitu 0.7,0.8, 0.9, 0.1 dan iterasi hingga 25. Kemudian spark secara otomatis membagi tiap dataset menjadi 3 secara random, terpisah dan tidak tumpang tindih. 3 data tersebut digunakan untuk train, validation dan test. Dengan rumus yang

dijabarkan pada persamaan pada (14) dan (15) mesin pembelajaran spark berusaha mencari model terbaik dari kombinasi parameter yang telah ditetapkan. Hasil dari proses training kemudian dapat dilihat pada Gambar 4.3.



Gambar 4.3 Hasil training dari Dataset 100K

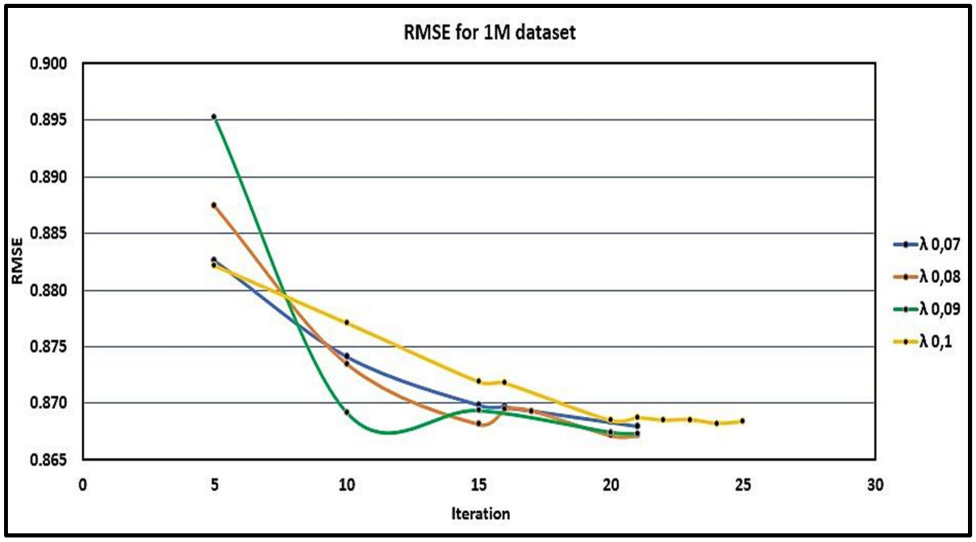
Dari gambar 4.3 terlihat bahwa dataset 1M yang miliki rating 1.000.209 berhasil dibagi menjadi tiga bagian yaitu 602.241 untuk training, 198.919 untuk validasi dan 199.049 rating yang digunakan untuk test. Best model yang didapatkan ada pada rank 12, lambda 0.1 dengan iterasi 20 mampu menghasilkan RMSE 0.86 pada validation dan 0.86 pada test set. Dapat dilihat bahwa hasil test dan validasi tidak overfitting karena RMSE antara validasi dan test tidak terlampau jauh.

Pengujian selanjutnya adalah peneliti ingin mengetahui sejauh mana terjadinya perubahan error terhadap parameter yang berbeda di setiap iterasi hingga mencapai titik konvergen. Peneliti kemudian menguji satu persatu paramater yaitu lambda dan rank pada dataset 1M. Hasil yang didapatkan pada dataset 1M dapat dilihat pada Tabel 4.4 dan Gambar 4.3.

Tabel 4.4 Hasil generate data 1M

Lambda	Iterasi	RMSE	Lambda	Iterasi	RMSE
0.07	5	0.883	0.09	5	0.895
	10	0.874		10	0.869
	15	0.87		15	0.869
	20	0.868		20	0.867
	21	0.868		21	0.867
0.08	5	0.887	0.1	5	0.882
	10	0.873		10	0.877
	15	0.868		15	0.872
	20	0.867		20	0.868
	21	0.867		21	0.869

Dari Tabel 4.4 terlihat bahwa paramater yang berbeda akan menghasilkan RMSE yang berbeda. Terlihat bahwa nilai lambda mempengaruhi hasil RMSE, semakin banyak iterasi data juga semakin terlatih yang menyebabkan RMSE menjadi semakin kecil namun ketika telah mencapai titik konvergen maka walaupun terus melakukan iterasi namun hasilnya tidak ada perubahan secara signifikan atau bahkan tidak berubah sama sekali. Terlihat bahwa best model berhasil didapatkan pada lambda 0.1 dan iterasi ke 20. Titik konvergen akan terlihat lebih jelas pada grafik yang ditunjukkan pada Gambar 4.4.



Gambar 4.4 Grafik RMSE untuk 1M dataset

Pada gambar 4.3 terlihat bahwa titik konvergen terjadi pada iterasi ke 20. Setelah iterasi ke 20, perubahan nilai RMSE terlihat tidak signifikan atau tidak berubah samasekali. Sementara hasil rekomendasi yang dihasilkan dapat dilihat pada Tabel 4.5. dan Tabel 4.6.

Tabel 4.5 Hasil TopN dari dataset 1M

Dataset	Ranking	Bobot	id movie	Title
1M	1	3,428	2858	American Beauty
	2	2,991	260	Star Wars: Episode IV - A New Hope
	3	2,990	1196	Star Wars: Episode V - The Empire Strikes Back
	4	2,883	1210	Star Wars: Episode VI - Return of the Jedi
	5	2,672	480	Jurassic Park
	6	2,653	2028	Saving Private Ryan
	7	2,649	589	Terminator 2: Judgment Day
	8	2,590	2571	Matrix, The
	9	2,583	1270	Back to the Future
	10	2,578	593	Silence of the Lambs, The

Tabel 4.6 Hasil Personal Rekomendasi dari dataset 1M

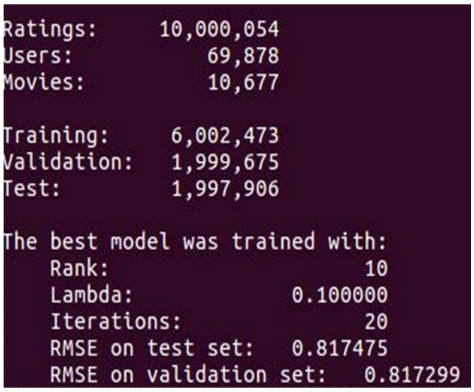
Input			Output	Validation
DataSet	**Title Movie**	**Rating**	**Title**	**RMSE**
1000000	Titanic	4	New Jersey Drive	Best Model = λ =0.1, Rank =10 15 Iteration RMSE = **0.868**
	Jurastic Park	2	Breakfast at Tiffanys	
	The Matrix	2	His Girl Friday	
	Toy Story	2	Just the Ticket	
	Home Alone	3	Ill Be Home For Christmas	
	City of Angles	4	Halloween 5: The Revenge of Michael Myers	
	Breaveheart	3	For the Moment	
	Starwars	2	Goya in Bordeaux (Goya en Bodeos)	
	Something to talk about	3	Message in a Bottle	
	Miracle on 34th street	3	Thomas and the Magic Railroad	

Tabel 4.5 adalah daftar rekomendasi yang diurutkan berdasarkan jumlah rating dari tiap film. Rating pada Top N ini adalah rating yang sebenarnya dan rating prediksi yang mengisi kolom matriks user-item. Seluruh rating tersebut di jumlahnya dan diperingkat menurut besanya jumlah rating. Berbeda dengan rekomendasi personal yang ditunjukkan pada tabel 4.6 yaitu rekomendasi untuk user yang telah

memasukkan rating film sebelumnya. Data dilihat kolom input dan kolom output. Pada pengujian mesin, peneliti mencoba memasukkan rating beberapa film yang disukai, lalu dengan best model, ALS-WR membuat rekomendasi personal dan menghasilkan rekomendasi film seperti yang bisa dilihat pada kolom output.

4.1.3 Pengujian Collaborative Filtering pada dataset 10M

Pengujian terhadap collaborative filtering dilakukan pada saat awal yaitu pada saat training. Pada dataset 10M diberikan parameter yaitu 10, 11 dan 12 dan lambda yaitu 0.7,0.8, 0.9, 0.1 dan iterasi hingga 25. Kemudian spark secara otomatis membagi tiap dataset menjadi 3 secara random, terpisah dan tidak tumpang tindih. 3 data tersebut digunakan untuk train, validation dan test. Dengan rumus yang dijabarkan pada persamaan pada (14) dan (15) mesin pembelajaran spark berusaha mencari model terbaik dari kombinasi parameter yang telah ditetapkan. Hasil dari proses training kemudian dapat dilihat pada Gambar 4.5.



Gambar 4.5 Hasil training dari Dataset 10M

Dari gambar 4.5 terlihat bahwa dataset 10M yang miliki rating 10.000.054 berhasil dibagi menjadi tiga bagian yaitu 6.002.473 untuk training, 1.999.675 untuk validasi dan 1.197.906 rating yang digunakan untuk test. Best model yang didapatkan ada pada rank 12, lambda 0.1 dengan iterasi 20 mampu menghasilkan RMSE 0.86 pada validation dan 0.86 pada test set. Dapat dilihat bahwa hasil test dan validasi tidak overfitting karena RMSE antara validasi dan test tidak terlampau jauh.

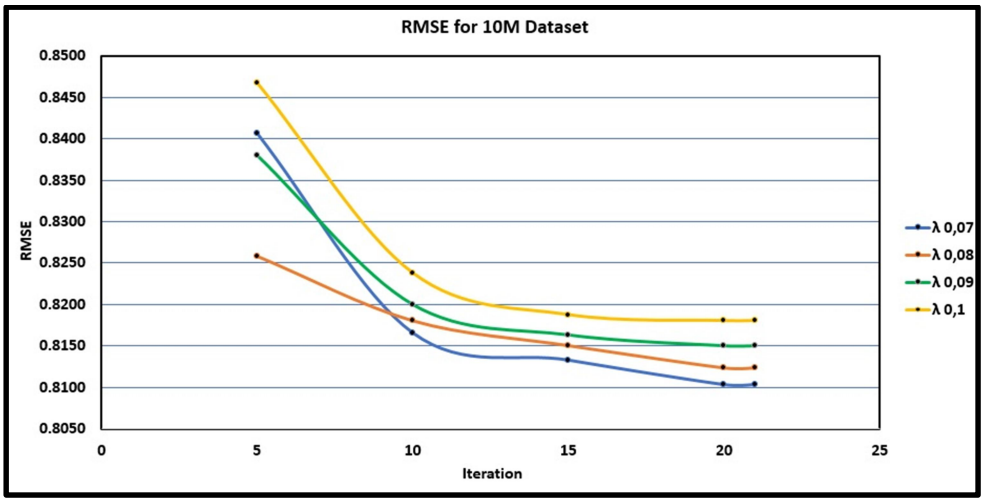
Pengujian selanjutnya adalah peneliti ingin mengetahui sejauh mana terjadinya perubahan error terhadap parameter yang berbeda di setiap iterasi hingga mencapai titik konvergen. Peneliti kemudian menguji satu persatu paramater yaitu

lambda dan rank pada dataset 1M. Hasil yang didapatkan pada dataset 1M dapat dilihat pada Tabel 4.7 dan Gambar 4.6.

Tabel 4.7 Hasil generate data 1M

Lambda	Iterasi	RMSE	Lambda	Iterasi	RMSE
0.07	5	0.8407	0.09	5	0.838
	10	0.817		10	0.820
	15	0.813		15	0.816
	20	0.810		20	0.815
	21	0.810		21	0.815
0.08	5	0.826	0.1	5	0.847
	10	0.818		10	0.824
	15	0.815		15	0.819
	20	0.812		20	0.818
	21	0.812		21	0.818

Dari Tabel 4.7 terlihat bahwa paramater yang berbeda akan menghasilkan RMSE yang berbeda. Terlihat bahwa nilai lambda mempengaruhi hasil RMSE, semakin banyak iterasi data juga semakin terlatih yang menyebabkan RMSE menjadi semakin kecil namun ketika telah mencapai titik konvergen maka walaupun terus melakukan iterasi namun hasilnya tidak ada perubahan secara signifikan atau bahkan tidak berubah sama sekali. Terlihat bahwa best model berhasil didapatkan pada lambda 0.1 dan iterasi ke 20. Titik konvergen akan terlihat lebih jelas pada grafik yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 Grafik RMSE untuk 10M dataset

Pada gambar 4.6 terlihat bahwa titik konvergen terjadi pada iterasi ke 20. Setelah iterasi ke 20, perubahan nilai RMSE terlihat tidak signifikan atau tidak berubah samasekali. Sementara hasil rekomendasi yang dihasilkan dapat dilihat pada Tabel 4.8. dan Tabel 4.9.

Tabel 4.8 Hasil TopN dari dataset 10M

Dataset	Ranking	Bobot	id movie	Title
10M	1	34,864	296	Pulp Fiction
	2	34,457	356	Forrest Gump
	3	33,668	593	Silence of the Lambs, The
	4	32,631	480	Jurassic Park
	5	31,126	318	Shawshank Redemption, The
	6	29,154	110	Braveheart
	7	28,951	457	Fugitive, The
	8	28,948	589	Terminator 2: Judgment Day
	9	28,566	260	Star Wars: Episode IV - A New Hope (a.k.a. Star Wars)
	10	27,035	150	Apollo 13

Tabel 4.9 Hasil Personal Rekomendasi dari dataset 10M

ALS-WR				
Input			Output	Validation
DataSet	Title Movie	Rating	Title	RMSE
10M	Titanic	4	From Beyond	Best Model = λ =0.1, Rank =10 20 Iteration RMSE = **0.818**
	Jurastic Park	2	Toy Story 2	
	The Matrix	2	Making the Grade	
	Toy Story	2	Honeymooners, The	
	Home Alone	3	Layer Cake	
	City of Angles	4	Unforgotten: Twenty-Five Years After Willowbrook	
	Breaveheart	3	Red Corner	
	Starwars	2	Surviving Picasso	
	Something to talk about	3	Living Sea, The	
	Miracle on 34th street	3	Dresser, The	

Tabel 4.8 adalah daftar rekomendasi yang diurutkan berdasarkan jumlah rating dari tiap film. Rating pada Top N ini adalah rating yang sebenarnya dan rating

prediksi yang mengisi kolom matriks user-item. Seluruh rating tersebut di jumlahnya dan diperingkat menurut besanya jumlah rating. Berbeda dengan rekomendasi personal yang ditunjukkan pada tabel 4.9 yaitu rekomendasi untuk user yang telah memasukkan rating film sebelumnya. Data dilihat kolom input dan kolom output. Pada pengujian mesin, peneliti mencoba memasukkan rating beberapa film yang disukai, lalu dengan best model, ALS-WR membuat rekomendasi personal dan menghasilkan rekomendasi film seperti yang bisa dilihat pada kolom output.

4.2 Performansi ALS-WR

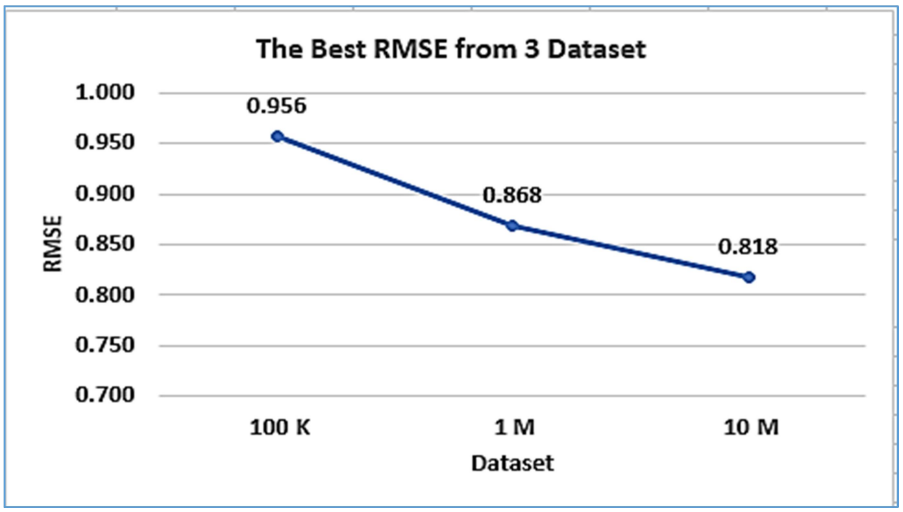
Pada metode memory-based dengan algoritma kNN terdapat kekurangan yaitu Skalabilitas dalam memory-based menjadi kelemahan dari model-based karena membutuhkan memori komputer yang besar karena memori diperlukan untuk proses seluruh database pada saat membuat prediksi. Sparsitas juga menjadi kelemahan dari memori based yaitu pada matriks user item yang besar, tidak semua terisi penuh, pengguna hanya merating beberapa item akibatnya terdapat matriks yang jarang (sparse), hal tersebut mengakibatkan ada beberapa pengguna aktif tidak mendapatkan prediksi karena pengguna aktif tidak memiliki barang yang sama dengan semua orang yang telah memberi nilai pada item target. Overfiting, dibutuhkan semua variabilitas acak dalam penilaian orang sebagai sebab akibat dan tidak ada pengujian. Dengan kata lain, algoritma berbasis memori tidak menggeneralisasi data sama sekali.

Dengan menggunakan algoritma ALS-WR overfiting dapat dihindari yang ditunjukkan pada dataset 100K hasil validasi menghasilkan RMSE 0.96 sementara hasil test adalah 0.94, pada dataset 1M nilai RMSE pada data validasi adalah 0.86 sementara RMSE pada dataset adalah 0.96, pada dataset 10M nilai RMSE data validasi adalah 0.81 sementara RMSE pada data test diperoleh 0.81. Dari datatersebut dapat dilihat bahwa algoritma ALS-WR dapat menanggulangi overfitting terbukti dari RMSE pada data set dan data validasi pada saat training hampir sama, hal tersebut juga dipengaruhi karena algoritma ALS-WR menggunakan paramater lambda sebagai regularisasi untuk mencegah overfiting, dan generalisasi terhadap data dilakukan dengan menggunakan iterasi yang berulang-ulang hingga mencapai titik konvergen seperti pada Gambar 4.2, Gambar 4.4 dan Gambar 4.6. Iterasi yaitu proses pengulangan peerhitungan dimana data dibagi menjadi tiga untuk

di training, divalidasi dan diteskan kembali hingga mencapai titik konvergen. Titik konvergen dicapai ketika kombinasi terbaik antara ranks, lambda berhasil dcapai yang ditunjukkan dengan RMSE terkecil. Kombinasi ranks dan lambda yang berbeda akan mempengaruhi error atau RMSE, hal ini yang membedakan antara metode memory based yang rentan akan overfiting dan model based yang diciptakan untuk memperbaiki dan mencegah overfiting data.

Sparsitas juga mampu dipecahkan oleh algoritma ALS-WR ini dimana terdapat fungsi untuk memprediksi missing-value. Dengan menggunakan best model, fungsi ke 5 dan ke 6 seperti yang dijelaskan pada bab 2 dijalankan yang menghasilkan prediksi rating seluruh item, pembobotan dan pengurutan ranking untuk keseluruhan item. Aktivitas ketiga diatas menghasilkan daftar Top N. Prediksi seluruh item tersebut kemudian disimpan dan digunakan kembali pada saat data baru masuk, jadi tidak seperti memori-based dimana pada saat data masuk menggunakan keseluruhan memori untuk memprediksi data, pada model-based, model fungsi dan prediksi data lama telah tersimpan sehingga untuk menghasilkan rekomendasi personal saat user baru memasuki sistem dengan rating yang baru, prosesnya tidak begitu berat dan memakan waktu lama.

Permasalahan skalabilitas juga berhasil dipecahkan, setelah training dilakukan, best model diciptakan untuk melakukan prediksi dan pemeringkatan, ketika data baru muncul, prosesnya hanya mencocokkan dan mencari rating yang mirip antar item masukan dengan item yang telah diprediksi, hal ini menguntungkan karena proses generate menjadi lebih cepat dan ringan dibandingkan melakukan semuanya dari awal seperti pada memori-based. RMSE yang dihasilkan ditunjukkan pada Gambar 4.7



Gambar 4.7 RMSE dari 3 dataset

Percobaan dengan mengenarate 3 dataset dengan size yang berbeda menghasilkan RMSE yang dirangkum pada 4.7 menunjukkan bahwa semakin besar data, justru error yang dihasilkan semakin kecil. Dengan demikian ALS-WR merupakan algoritma yang adaptif yang dapat digunakan pada data yang terus tumbuh atau big data.

Dari hasil rekomendasi ketiga dataset dapat dilihat pula bahwa hasil rekomendasi yang didapatkan berbeda dengan *search engine*. Dengan masukan rating baru untuk 5 item, didapatkan hasil bahwa judul item yang direkomendasikan berbeda dengan judul item yang dirating. Dengan demikian *serendipity problem* yang menjadi keterbatasan dari *search engine* berhasil juga dipecahkan melalui algoritma ALS-WR ini.

4.3 Pengujian Genre Similiarity

Genre dari hasil collaborative filtering cendrung random, tidak mengikuti genre movie yang diinputkan oleh karena itu pendekatan kemiripan diperlukan agar presisi model ini menjadi tinggi, ini mengakomodasi user yang lebih tertarik movie berdasarkan genre dibandingkan saran dari pengguna lain yang memiliki selera yang sama.

Tabel 4.10 Hasil Perhitungan cosine similarity

Input	Output	
Title & Genre	**Title Movie**	**Cosine Value**
(Data set 100K) Low Down Dirty Shame, A : Action Comedy	Low Down Dirty Shame	1
	D E B S	1
	Tuxedo	1
	Hard Way	1
	The Pacifier	1
	Mr.Nice Guy	1
	I love Trouble	1
	Taxi 3	1
	Talladega Nights: The Ballad of Ricky Roby	1
	Three Fugitives	1
(Data set 1M) New Jersey Drive: Crime,Drama	New Jersey Drive	1
	Piece of the Action	1
	Dead man walking	1
	Best Laid Pians	1
	insquistor, the	1
	Cariltas Way	1
	Drighstore Cowboy	1
	Hoax	1
	City by the sea	1
	Baby Boy	1
(Data set 10M) Red Corner : Crime, Thriller	Red corner	1
	Bank Job,the	1
	16 Blocks	1
	Murder by number	1
	Night Visitor	1
	Postman Always Rings Twice, the	1

Untuk itu pada rekomendasi kedua, hasil dari algoritma ALS-WR yang telah diurutkan, diambil 1000 urutan teratas untuk di dekatkan kembali menurut genre nya dengan menggunakan algoritma cosine similiarity. Hasil dari perhitungan cosine similarity dapat dilihat pada Tabel 4.10.

Sebagai contoh film Jurastic Park bergenre :Action|Adventure|Sci-Fi|Thriller menghasilkan rekomendasi dengan judul Spiderman yang bergenre Action|Adventure|Sci-Fi|Thriller bernilai cosine 1, dari 1000film yang telah diseleksi hanya menghasilkan 3 film yang bernilai 1 yang berarti memiliki kesamaan genre 100% dengan genre film Juristic Park, film lainnya seperti The Cave (2005) memiliki nilai cosine 0.894 karena memiliki genre Action|Adventure|Horror|Sci-Fi|Thriller walau ada 4 genre sama tetapi ada tambahan genre horror pada film tersebut yang mengakibatkan penurunan nilai cosine begitu juga pada film Godzilla(1998) memiliki genre Action|Sci-Fi|Thriller memiliki kekurangan genre Adventure yang menyebabkan genre dinilai tidak sama 100% dengan genre film Jurasticpark dan itu menyebabkan nilai cosine menjadi turun.

Tabel 4.11 Film dengan genre tidak mirip 100%

No.	Judul Film	Nilai Cosine
1	Jurastic Park (1993	1
2	Spider-Man (2002)	1
3	Babylon A.D (2008)	1
4	Cave,The (2005)	0.894
5	Ballistic:Ecks vs Server (2002)	0.866
6	Paycheck (2003)	0.866
7	Godzilla (1998)	0.866
8	Red Planet (2000)	0.866
9	Hangar 18 (1980)	0.866
10	Quantum of Solace (2008)	0.866

Kemudian dari hasil *cosine similiarity* bisa dilihat nilai presisinya sebagai pada Tabel 4.11. Dari Tabel 4.11 dapat dilihat bahwa peneliti hanya menampilkan 10 movie di halaman interface pengguna, dari query yang diinputkan dimana yang diambil adalah genrenya saja, terlihat 10 movie yang terbaik bernilai 1 (satu) itu berarti kesepuluh movie tadi mempunyai tingkat kemiripan 100% dengan query, semakin dinaikkan thresholdnya, maka movie yang ditemukan akan semakin sedikit, nilai presisinya juga akan semakin sedikit, namun dengan membatasi jumlah movie yang ditampilkan hanya 10 movie, ini akan memberikan rekomendasi terbaik untuk

pengguna bahwa hanya movie yang memiliki tingkat kemiripan tertinggi saja yang diberikan kepada pengguna.

Tabel 4.11 Hasil Cosine Similiarity dan Evaluasi Presisi

Input	Output				
Title & Genre	**Title Movie**	**Cosine Value**	**Thrshold**	**Relevant Movie**	**Precission**
(Data set 100K) Low Down Dirty Shame, A : Action Comedy	Low Down Dirty Shame	1	10%	285	0.95
	D E B S	1	20%	285	0.95
	Tuxedo	1	30%	285	0.94
	Hard Way	1	40%	283	0.77
	The Pacifier	1	50%	233	0.54
	Mr.Nice Guy	1	60%	163	0.41
	I love Trouble	1	70%	126	0.41
	Taxi 3	1	80%	16	0.05
	Talladega Nights: The Ballad of Ricky Roby	1	90%	13	0.04
	Three Fugitives	1	100%	13	0.04
(Data set 1M) New Jersey Drive: Crime,Drama	New Jersey Drive	1	10%	285	0.94
	Piece of the Action	1	20%	285	0.94
	Dead man walking	1	30%	284	0.94
	Best Laid Pians	1	40%	261	0.86
	insquistor, the	1	50%	222	0.73
	Cariltas Way	1	60%	166	0.55
	Drighstore Cowboy	1	70%	166	0.55
	Hoax	1	80%	20	0.06
	City by the sea	1	90%	17	0.05
	Baby Boy	1	100%	17	0.05
(Data set 10M) Red Corner : Crime, Thriller	Red corner	1	10%	278	0.92
	Bank Job,the	1	20%	278	0.92
	16 Blocks	1	30%	277	0.92
	Murder by number	1	40%	245	0.81
	Night Visitor	1	50%	159	0.52
	Postman Always Rings Twice, the	1	60%	72	0.23
	Day of Jackal, The	1	70%	71	0.23
	Night Moves	1	80%	39	0.12
	Nightcap	1	90%	12	0.03
	Internal Affairs	1	100%	12	0.03

4.4 Performansi *Cosine Similiarity*

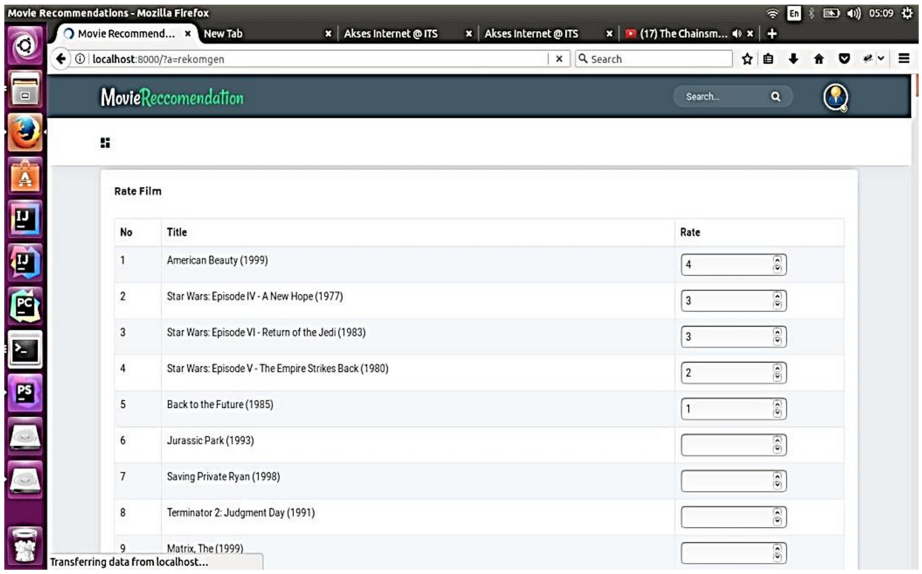
Seperti yang digambarkan Tabel.2 bahwa hasil dari collaborative filtering sangat tidak terduga, beberapa pengguna terkadang menyukai genre tertentu, oleh karena itu hasil dari collaborative filtering difilter kembali berdasarkan kemiripan genre nya dengan menggunakan cosine similarity. Hasil presisi yang didapatkan dengan menggunakan cosine similarity dapat dilihat pada Tabel.4 bahwa dengan hanya menampilkan 10 item film di halaman interface pengguna, dari query yang

diinputkan dimana yang diambil adalah genrenya saja, terlihat 10 yang terbaik bernilai 1 (satu) itu berarti kesepuluh item tadi mempunyai tingkat kemiripan 100% dengan query, semakin dinaikkan thresholdnya, maka item yang relevan berdasarkan item film pilihan pengguna, ditemukan akan semakin sedikit, nilai presisinya juga akan semakin kecil, namun dengan membatasi jumlah item yang ditampilkan hanya 10 item, ini akan memberikan rekomendasi terbaik untuk pengguna bahwa hanya item yang memiliki tingkat kemiripan tertinggi saja yang diberikan kepada pengguna. Dengan demikian, menggunakan 2 algoritma ALS-WR dan cosine similiarity tadi pengguna bisa mendapatkan hasil terbaik dengan error kecil dan presisi yang tinggi.

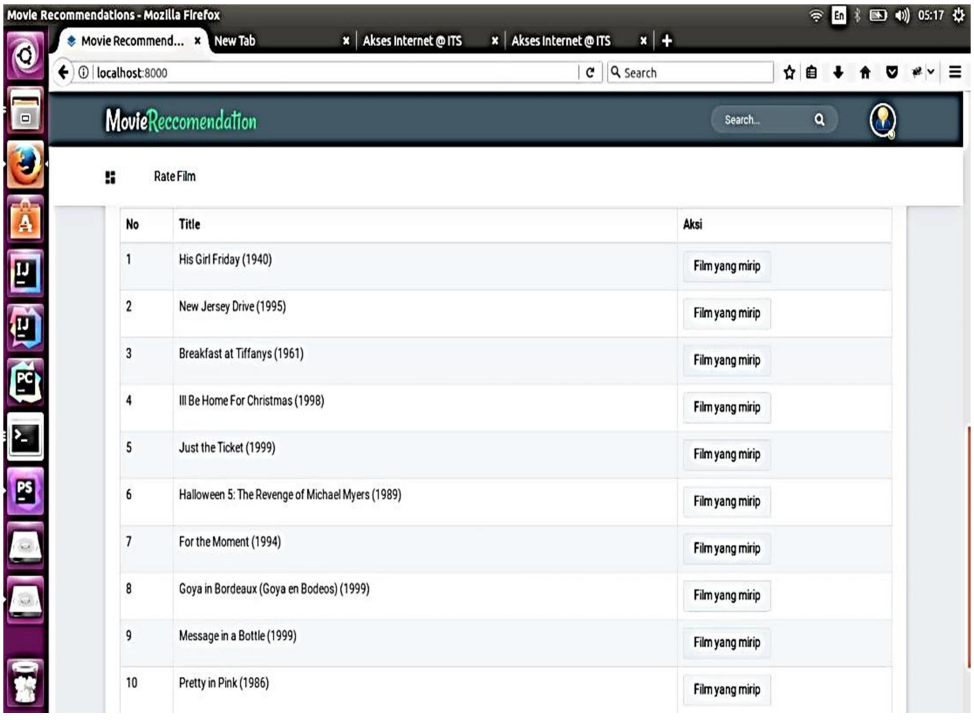
4.5 Uji Penerimaan User

Uji Penerimaan User yaitu dengan menguji hasil rekomendasi kepada pengguna yang mencoba mesin rekomendasi ini. Interface dari mesin perlu dihadirkan agar user dapat dengan mudah memberikan rating dan mendapatkan hasil rekomendasi. Interface dan perhitungan cosine similiarity menggunakan phpstorm dengan bahasa php. Terlebih dahulu engine rekomender yang dibuat yaitu untuk menjalankan ALS-WR. Spark dan mlib spark terlebih dahulu dijalankan pertamakali di terminal, setelah Top N recommendation berhasil didapatkan kemudian disimpan di SQL, karena Spark sendiri memiliki API yang dapat terhubung dengan SQL.

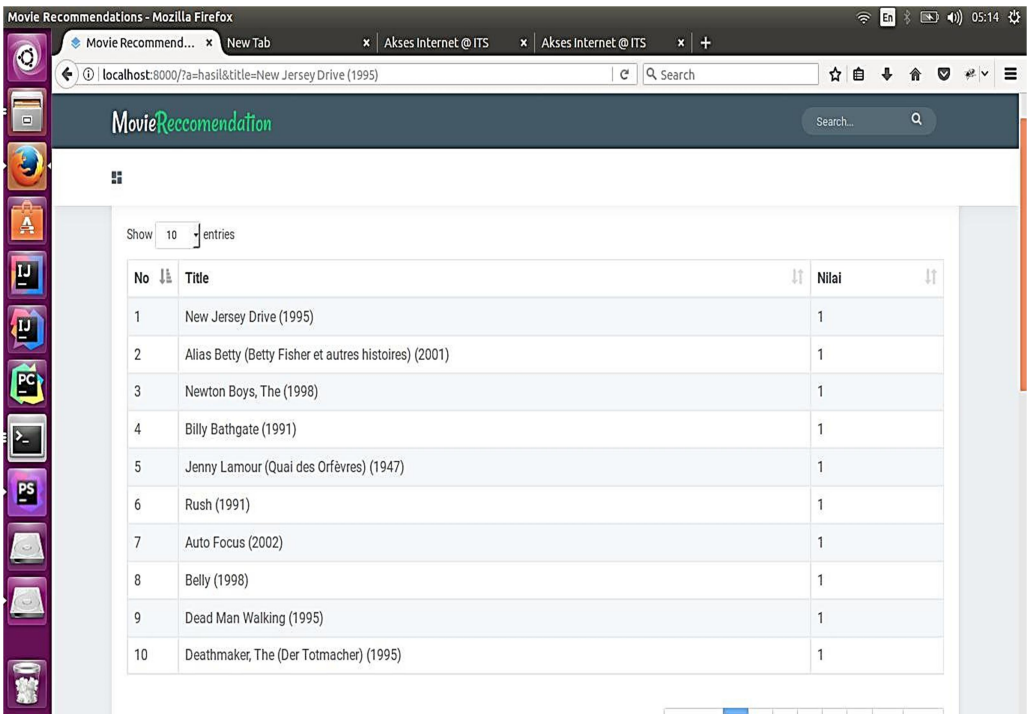
PHP dapat terhubung dengan Mlib Spark, jadi setelah user selesai me rating dan mengklik tombol “run” secara otomatis spark berjalan untuk memproses rekomendasi I dengan algoritma ALS-WR. Setelah daftar rekomendasi I keluar, user dapat mengklik salah satu judul film kemudian proses cosine similiarity berjalan kemudian daftar rekomendasi II dapat keluar. Dibawah ini adalah gambar interface sederhana agar user dapat merating melalui web.



Gambar 4.8 Halaman Input Rating



Gambar 4.9 Halaman Hasil Rekomendasi pertama



Gambar 4.10 Halaman Rekomendasi kedua

Untuk *collaborative filtering* pada data set 1M dipasang paramater best model dengan rank 10, lambda 0,8 dan untuk cosine similarity diberikan batas threshold 50% atau 0,5. Kemudian hasil rekomendasi dari parameter yang telah di set ini diberikan kepada 20 orang user. User mencoba memberikan rating untuk item film yang pernah dia sukai atau pernah ditonton, kemudian diberikan kuesioner yang terdiri dari 5 pertanyaan yang dapat dilihat pada Bab 3 dan lampiran. Hasil nya dapat dilihat pada Tabel 4.12.

Tabel 4.12 Hasil Kuesioner

No.	Daftar Pertanyaan	Jawaban	
1	Dari daftar rekomendasi pertama, apakah memberikan hasil yang tidak terduga (mengejutkan) yang anda tidak menyangka itu ada ternyata film tersebut tersedia ?	Ya 90%	Tidak 10%
2.	Dari 10 daftar Rekomendasi I, berapa jumlah rekomendasi yang relevan menurut selera anda ?	Rata-rata Presisi : 28%	
3.	Klik salah satu dari 10 daftar yang dihasilkan dari rekomendasi metode I, kemudian lihat 10 daftar rekomendasi selanjutnya. Apakah memberikan hasil yang tidak terduga ?	Ya 70%	Tidak 30%
4.	Dari 10 daftar rekomendasi yang tersedia pada lembar rekomendasi II, berapa jumlah rekomendasi yang relevan menurut selera Anda ?	Rata-rata Presisi : 62%	
5.	Rekomendasi yang manakah yang lebih baik menurut selera Anda diantara kedua hasil rekomendasi tersebut ?	Rekomendasi I 25%	Rekomendasi II 75%

Dari Tabel 4.12 dapat dilihat bahwa hasil rekomendasi pertama dari collaborative filtering memberikan dampak yang mengejutkan bagi 90% responden akan tetapi dari 10 item film yang direkomendasikan, hanya 28% yang membuat user tertarik untuk melihat lebih jauh tentang item film tersebut. Setelah item film difilter kembali menurut genrenya, tingkat peminatan terhadap item film yang direkomendasikan meningkat menjadi 62% dari rata-rata seluruh responden, namun karena item film bergenre hampir sama memberikan efek tidak begitu mengejutkan. Hasil akhir ternyata 75% responden lebih menyukai rekomendasi kedua yaitu hasil dari filtering dua tahap dibandingkan hanya collaborative filtering saja.

BAB 5
KESIMPULAN

5.1 Kesimpulan

Mesin rekomendasi untuk data yang terus tumbuh hingga menggunakan metode collaborative filtering dengan algoritma ALS-WR dalam memprediksi rating untuk semua item menghasilkan RMSE 0,89 untuk dataset 100K, 0,86 untuk dataset 1M dan 0,81 untuk dataset 10M ini menggambarkan bahwa ALS-WR dapat dapat mengatasi masalah skalabilitas untuk data yang terus tumbuh. Hal tersebut ditunjukkan dengan semakin besar data, tingkat error justru semakin kecil..

ALS-WR juga dapat mengatasi sparsitas dalam memberikan prediksi rating seluruh item. Dengan menggunakan best model yang dijalankan yang menghasilkan prediksi rating seluruh item serta pembobotan dan pengurutan ranking untuk keseluruhan item. Aktivitas ketiga diatas menghasilkan daftar Top N. Prediksi seluruh item tersebut kemudian disimpan dan digunakan kembali pada saat data baru masuk, jadi tidak seperti memori-based dimana pada saat data masuk menggunakan keseluruhan memori untuk memprediksi data, pada model-based yang menggyunakan ALS-WR, best model telah tersimpan sehingga untuk menghasilkan rekomendasi personal saat user baru memasuki sistem dengan rating yang baru, prosesnya tidak berat dan memakan waktu lama.

Dengan menggunakan algoritma ALS-WR overfiting dapat dihindari yang ditunjukkan pada dataset 100K hasil validasi menghasilkan RMSE 0.96 sementara hasil test adalah 0.94, pada dataset 1M nilai RMSE pada data validasi adalah 0.86 sementara RMSE pada dataset adalah 0.96, pada dataset 10M nilai RMSE data validasi adalah 0.81 sementara RMSE pada data test diperoleh 0.81. Dari data tersebut dapat dilihat bahwa algoritma ALS-WR dapat menanggulangi overfitting terbukti dari RMSE pada data set dan data validasi pada saat training hampir sama.

Dengan menggunakan metode 2 tahap filtering ini item dapat diperingkat menurut prediksi rating yang dihasilkan dari faktorisasi keseluruhan rating namun tetap dapat didekatkan dengan genre yang dipilih user. Untuk penilaian penerimaan user terhadap hasil rekomendasi yang diberikan melalui metode 2 tahap ini, tingkat penerimaan terhadap item film yang direkomendasikan meningkat dari rata-rata 28%

untuk rekomendasi dengan collaborative filtering menjadi 62% untuk rekomendasi dengan metode kemiripan genre berbasis collaborative filtering (2 tahap filtering). Hasil akhir ternyata 75% responden lebih menyukai rekomendasi kedua yaitu hasil dari dua tahap filtering dibandingkan hanya collaborative filtering saja.

5.2 Saran

Pada penelitian selanjutnya diharapkan menggunakan kombinasi algoritma yang berbeda pada matriks faktorisasi, misalnya dengan menggunakan matriks faktorisasi dengan penambahan karnelisasi dan naive bayes untuk menghitung probabilitas kemiripan genre, kemudian dapat diperbandingkan tingkat presisinya dari model ini.

DAFTAR PUSTAKA

[1] D. Price. (2015, February 2017). *Suprising Facts and Stats about The Big Data Industry*. Available: http://cloudtweaks.com/2015/03/surprising-facts-and-stats-about-the-big-data-industry/

[2] c. Anderson. (2008). *The Long Tail: Why the Future of Business is Selling Less of More*.

[3] D. Asanov, "Algorithms and Methods in Recommender Systems.," *International Journal of Computer Applications,* vol. 118, 2015.

[4] M. RIDWAN. (Maret 2017). *Predicting Likes: Inside A Simple Recommendation Engine's Algorithms*. Available: https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine

[5] J. M. O'Brien. (2006, 14 January). *The race to create a 'smart' Google*. Available: http://archive.fortune.com/magazines/fortune/fortune_archive/2006/11/27/8394347/index.htm

[6] (2012). *Netflix Yields $131 Value With User Recommendation Tools*. Available: http://www.forbes.com/sites/greatspeculations/2012/04/17/netflixs-yields-131-value-with-user-recommendation-tools/#5b5a177f199a.

[7] B. Morrison. (2016, 18 February). *What Do Google, Netflix, Amazon and Best Buy Have In Common?* Available: https://www.nectarom.com/google-netflix-amazon-best-buy-common/.

[8] J. Roettgers. (2014, 12 Maret 2017). *Netflix spends $150 million on content recommendations every year*. Available: https://gigaom.com/2014/10/09/netflix-spends-150-million-on-content-recommendations-every-year/

[9] *Creative Marketing for New Product and New Business Development* Singapore: World Scientific Publishing, 2008.

[10] L. R. F Ricci, B Shaphira. (2011). *Recommender Systems Handbook*.

[11] M. R. a. A. Walker, "Supporting 'word of mouth Social Networks through Collaborative Filtering," *Journal of Interactive Learning Research,* vol. 14, pp. 78-79, 2003.

[12] X. S. a. T. M. Khoshgoftar, "A Survey of Collaborative Filtering Techniques," *Advances in Artivicial Inteliigence Journal,* vol. 2009, 2009.

[13] F. S. J Bobadilla, J. Bernal "A New Collaborative Filtering Maetric That improves the behavior of Recommender Systems," *Knowledge-Based Systems Journal,* vol. 23, pp. 520-528, 2010.

[14] G. K. Badrul Sarwar, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *international conference on World Wide Web*, Hongkong, 2001, pp. 285-295.

[15] S. K. T. a. S. K. Shrivastava, "An Approach for Recommender System by Combining Collaborative Filtering with User Demographics and Items Genres," *International Journal of Computer Applications,* vol. 128, 2015.

[16] A. G. a. A. Prugel-Bannett, "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering," in *International Multi Conferenceof Engineers and Computer Science*, Hongkong, 2010.

[17] O. Fiedler. (2014, 17 February). *Machine Learning at Scale*. Available: https://github.com/OndraFiedler/spark-recommender/blob/master/reportAndDocumentation.pdf

[18] C. R. Aberger, "Recommender: An Analysis of Collaborative Filtering Techniques," Stanford University, California2014.
[19] D. W. Yunhong Zhou, Robert Schreiber and Rong Pan, "Large-scale Parallel Collaborative Filtering for the Netflix Prize. Proceeding " in *4th international conference on Algorithmic Aspects in Information and Management*, 2008, pp. 337 – 34.
[20] A. Sanjung, "Perbandingaan Semantic Classification dan Cluster-based Smoothed pada Recommender System berbasis Collaborative filtering.," S2, Teknik Informatika, Telkom University, Bandung, 2011.
[21] A. Handrico, "Sistem Rekomendas Buku Perustakaan Fakultas Sains dan Teknologi Dengan Metode Collaborative Filtering," S1, Teknik Informatika, Universitas Islam Riau, Riau, 2012.
[22] Z. M. a. F. D Jannach. (2011). *Recommender System – An Introduction*.
[23] M. S. Gawesh Jawaheer, Patty Kostkova, "Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service," in *International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, Barcelona, Spain, 2010, pp. 47-51
[24] N. Pantreath. (2015). *Machine Learning with Spark*.
[25] I. A. N. Hakim, "Sistem Rekomendasi Film Berbasis Web Menggunakan Metode Item-Based Collaborative Filtering Berbasis K-Nearest Neighbor," S1, Program Studi Ilmu Komputer, Universitas Pendidikan Indonesia, Bandung, 2010.
[26] S. Marafi. (2014, April 2017). *Collaborative Filtering with R*. Available: Collaborative Filtering with R
[27] C. McDonald. (2015, 26 April). *Pararel and Iterative Processing for Machine Learning Recommendation With Spark*. Available: https://mapr.com/blog/parallel-and-iterative-processing-machine-learning-recommendations-spark/
[28] R. Burke, "Hybrid Web Recommender Systems," in *The Adaptive Web*, A. K. P. Brusilovsky, and W. Nejdl Ed., ed. Berlin: Springer, 2007, pp. 377 – 408.
[29] R. M. a. R. N. Prem Melville, "Content-boosted collaborative filtering for improved recommendations," in *Artificial intelligence*, Menlo Park, CA, USA, 2002, pp. Pages 187-192
[30] B. Atmaja, "Content-Boosted Collaborative Filtering Pada Recommender System," S1, Teknik Informatika, Telkom University, Bandung, 2011.
[31] Y. Z. a. W. Song, "A Collaborative Filtering Recommendation Algorithm Based on Item Genre and Rating Similarity," presented at the IEEE conference :International Conference on Computational Intelligence and Natural Computing, 2009.
[32] S.-K. K. Sang-Min Choi, Yo-Sub Han, "A movie recommendation algorithm based on genre correlations," *International Journal Expert Systems with Applications,* vol. 39, pp. 8079-8085 Jully 2012 2012.
[33] C. Eaton, D. Dirk, D. Tom, L. George, and Z. Paul, *Understanding Big Data*: Mc Graw Hill.
[34] E. Dumbill, *Big Data Now Current Perspective*: O'Reilly Media, 2012.
[35] S. Connolly, "Open Thoughts on Software, Business, and Life," in *7 Key Drivers for the Big Data Market*, ed, 2012.
[36] P. S. V Srinivas Jonnalagadda, Krishnamachari Thumati "A Review Study of Apache Spark in Big Data Processing," *International Journal of Computer Science Trends and Technology (IJCST),* vol. 4, pp. 93-98, May-Jun 2016 2016.

[37] C. S. W. Widayati, "KOMPARASI BEBERAPA METODE ESTIMASI KESALAHAN PENGUKURAN," *Jurnal Penelitian dan Evaluasi Pendidikan,* vol. 2, 2009.
[38] J. Han. (2012). *Data mining : Concepts and Techniques*.
[39] J. A. K. Badrul M. Sarwar, Al Borcher, Jon Herlocker, Brad Miller, and John Riedl, "Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System," in *Computer supported cooperative work*, Seattle, Washington, USA, 1998, pp. 345-354
[40] A. G. a. G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research,* vol. 10, 2009.

Halaman ini sengaja dikosongkan

LAMPIRAN I

KUESIONER

Uji Penerimaan User Terhadap Hasil Rekomendasi

Nama :

Alamat :

No.	Daftar Pertanyaan	Jawaban	
1	Dari daftar rekomendasi pertama, apakah memberikan hasil yang tidak terduga (mengejutkan) yang anda tidak menyangka itu ada ternyata film tersebut tersedia ?	Ya	Tidak
2.	Dari 10 daftar Rekomendasi I, berapa jumlah rekomendasi yang relevan menurut selera anda ?		
3.	Klik salah satu dari 10 daftar yang dihasilkan dari rekomendasi metode I, kemudian lihat 10 daftar rekomendasi selanjutnya. Apakah memberikan hasil yang tidak terduga ?	Ya	Tidak
4.	Dari 10 daftar rekomendasi yang tersedia pada lembar rekomendasi II, berapa jumlah rekomendasi yang relevan menurut selera Anda ?	Rata-rata Presisi :	
5.	Rekomendasi yang manakah yang lebih baik menurut selera Anda diantara kedua hasil rekomendasi tersebut ?	Rekomendasi I	Rekomendasi II

Tanggal :

Tandatangan :

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS

Penulis lahir di Jakarta pada tanggal 4 Januari 1984, yang merupakan putri sulung dari tiga bersaudara. Penulis pernah menempuh pendidikan dasar di SDN Curug V, lalu pendidikan menengah di SMPN 1 Cimanggis, dan pendidikan menengah atas di SMUN 98 Jakarta. Kemudian melanjutkan ke jenjang perguruan tinggi S1 di Jurusan Ilmu Perpustakaan dan Informasi, Fakultas Ilmu Pengetahuan Budaya, Universitas Indonesia (UI) lulus pada tahun 2007 dengan judul Skripsi : Miriam Budiardjo Resource Center dan American Corner dalam Pengertian Propaganda.

Pengalaman kerja antara lain sebagai staf Pusat Data dan Informasi Komnas HAM tahun 2007-2009. Dan pada tahun 2009 hingga saat ini penulis bekerja sebagai PNS pada Sekolah Tinggi Energi dan Sumber Daya Mineral (STEM), Kementerian Energi dan Sumber Daya Mineral (KESDM) dengan jabatan fungsional sebagai Pustakawan. Pada tahun 2015 mengikuti program beasiswa pendidikan S2 dari Kementerian Komunikasi dan Informasi (Kominfo) pada Bidang keahlian Telematika – CIO (Chief Information Officer) di Teknik Elektro ITS Surabaya.

Penulis dapat dihubungi melalui email : indahsurvyana@gmail.com