



UNIVERSITAS INDONESIA

**ANALISA KINERJA
METODE FAKTORISASI MATRIKS PROBABILITAS
PADA SISTEM REKOMENDASI**

TESIS

**UMAR RUSWANDI
1006786322**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM MAGISTER MATEMATIKA
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**ANALISA KINERJA
METODE FAKTORISASI MATRIKS PROBABILITAS
PADA SISTEM REKOMENDASI**

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Sains

**UMAR RUSWANDI
1006786322**

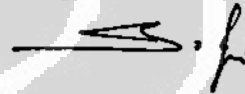
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM MAGISTER MATEMATIKA
DEPOK
JUNI 2012**

HALAMAN PERSETUJUAN

Judul : Analisa Kinerja Metode Faktorisasi Matriks Probabilitas
pada Sistem Rekomendasi
Nama : Umar Ruswandi
NPM : 1006786322

Tesis ini telah diperiksa dan disetujui

Depok, 25 Juni 2012

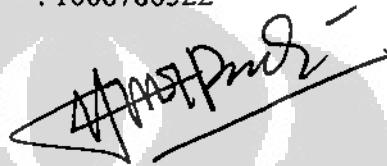


Dr. rer. nat. Hendri Murfi, M.Kom
Pembimbing Tesis

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Umar Ruswandi
NPM : 1006786322



Tanda Tangan : _____
Tanggal : 25 Juni 2012

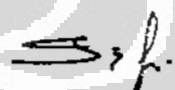
HALAMAN PENGESAHAN

Tesis ini diajukan oleh:

Nama : Umar Ruswandi
NPM : 1006786322
Program Studi : Magister Matematika
Judul Tesis : Analisa Kinerja Metode Faktorisasi Matriks Probabilitas
pada Sistem Rekomendasi

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Sains pada Program Studi Magister Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. rer. nat. Hendri Murfi, M.Kom. (.....)

Penguji : Dr. Kiki Ariyanti S. (.....)

Penguji : Dr. Dian Lestari (.....)

Penguji : Dr. Hengki Tasman M.Si (.....)

Ditetapkan di : Depok

Tanggal : 25 Juni 2012

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan kekuatan sehingga penulis dapat menyelesaikan penulisan tesis ini. Penyusunan tesis ini sebagai salah satu persyaratan memperoleh gelar Magister Sains pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia. Penulis sangat menyadari terdapat berbagai kekurangan dalam penulisan tesis ini, karena terbatasnya waktu, kemampuan yang dimiliki, sehingga penulisan tesis ini masih jauh dari sempurna.

Akhirnya pada kesempatan ini penulis mengucapkan terima kasih yang sedalam-dalamnya kepada berbagai pihak yang telah membantu dalam penulisan tesis ini, khususnya kepada :

1. Fakultas Psikologi UI yang telah memberikan kesempatan kepada penulis untuk menempuh pendidikan pada Program Studi Magister Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia
2. Bapak Dr. rer. nat Hendri Murfi selaku dosen pembimbing atas segala kesabarannya dalam membimbing penulis dalam menyelesaikan tesis ini.
3. Bapak Prof. Dr. Djati Kerami selaku Ketua Program Studi Magister Matematika dan Dr. Yudi Satria, M.T selaku Ketua Departemen Matematika yang telah memberikan masukan yang sangat berharga selama penulis menjalani pendidikan.
4. Para penguji tesis, Dr. Kiki Arianti S, yang telah memberikan catatan kritis dan koreksi yang sangat berharga dalam penulisan tesis. Dr. Dian Lestari yang telah banyak memberikan pemahaman statistik yang baik pada penulis dan Dr. Hengki Tasman selaku dosen aljabar linier lanjut yang telah mengilhami penulis dalam memahami proses pembuktian teorema-teorema matematika.
5. Seluruh Dosen dan karyawan Magister Matematika yang telah banyak membantu penulis selama menjalani pendidikan matematika di FMIPA UI.

6. Istri dan putraku tercinta atas segala doa, perhatian, kesabaran dan dukungan selama penulis menempuh pendidikan.
7. Prof. Dr. M. Enoch Markum selaku Ketua Lembaga Penelitian Psikologi UI dan Dr. Siti Purwanti B selaku Manajer Umum Fakultas Psikologi UI yang telah mendorong penulis untuk melanjutkan studi ke jenjang magister.
8. Rekan-rekan di Unit IT Fakultas Psikologi : Pak Tatan, Mas Gugah, Pak Nasori, Syukron dan Rachmad yang telah memberi dukungan selama penulis menjalani pendidikan .
9. Alhadi Bustaman, Ph.D selaku dosen matakuliah Matriks yang telah memberi kesempatan kepada penulis dalam mengeksplorasi teknik komputasi sistem paralel.
10. Sony Pawoko MTI selaku Koordinator TI Perpustakaan Pusat yang telah memberikan masukan yang sangat berharga dalam redaksi penulisan tesis.
11. Rekan-rekan kelompok diskusi *machine learning* : Supri, Subian, Dila, Rina, Endaryono, Setiawan, dan Desti terima kasih atas kebersamaanya.
12. Fathin, Gita, Siti, Deby, Peter John, Dahlia, Sigit, Pak Tarhadi, Feni, Uun, Rifki, Ifa, Tri, Haryono, Mujayin, Huda, Haryono, Ningrum, Bob dan rekan-rekan Magister Matematika angkatan 2010 yang tidak dapat saya sebutkan satu-persatu . Terima kasih atas kebersamaanya selama ini.

Depok, 25 Juni 2012

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertandatangan di bawah ini:

Nama : Umar Ruswandi
NPM : 1006786322
Program Studi : Magister Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Tesis

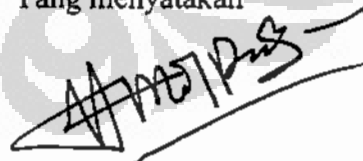
demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul:

Analisa Kinerja Metode Faktorisasi Matriks Probabilitas pada Sistem
Rekomendasi

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 25 Juni 2012
Yang menyatakan



(Umar Ruswandi)

ABSTRAK

Nama : Umar Ruswandi
Program Studi : Magister Matematika
Judul : Analisa Kinerja Metode Faktorisasi Matriks Probabilitas pada Sistem Rekomendasi

Tujuan dari penelitian ini adalah menguji akurasi dari metode faktorisasi matriks probabilitas (pmf) pada data rating film MovieLens. Dalam pmf, data ini dapat direpresentasikan dalam bentuk matriks R yang berukuran $n \times m$, dimana n adalah banyaknya pengguna sedangkan m adalah banyaknya judul film. Keluaran dari model ini adalah berupa dua buah matriks W dan H . Dimana W adalah matriks fitur pengguna sedangkan H adalah matriks fitur film. Akurasi dari model tercermin dari besarnya norm Frobenius $\|R-WH\|$ pada data testing. Matriks W dan H dapat diestimasi dengan menggunakan Teorema Bayes. Berdasarkan Teorema ini, model yang baik adalah model yang memiliki probabilitas posterior maksimum. Dari eksperimen, kondisi tersebut dicapai saat parameter $k=17$ dan $\lambda=0.2$ dengan $RMSE=0.920661$. Pada nilai $RMSE$ tersebut model ini masuk kategori yang baik dalam memprediksi banyaknya genre dan skor kosong dalam matriks R .

Kata kunci : Sistem Rekomendasi, Faktorisasi Matriks, *Collaborative Filtering*, Akurasi, *Social Network Analysis*, Reduksi Dimensi
xviii+59 halaman : 9 gambar; 12 tabel, 1 Algoritma
Daftar Pustaka : 45 (1992-2012)

ABSTRACT

Name : Umar Ruswandi
Program Study : Magister of Matematic
Title : Performance Analysis of Probabilistic Matrix Factorization
Method on Recommendation System

The purpose of this study is to test the accuracy of the method of probabilistic matrix factorization (PMF) on MovieLens movie rating data. In PMF, this data can be represents by the sparse matrix R . size $n \times m$, where n is the number of users, while m is the number of movie titles. The output of the model is in the form of two matrices W and H . Where W is the matrix of user features, while H is the matrix of films features. The accuracy of the model is reflected in the size of the Frobenius norm $\|R - WH\|$ in the data testing. Matrices W and H can be estimated by using Bayes theorem. Based on this theorem, a good model is a model that has a maximum posterior probability. From these experiments, the condition is achieved when the parameters $k = 17$ and $\lambda = 0.2$ with $RMSE = 0.920661$. In this model, the RMSE values in the category of good in predicting the number of genre and empty scores in the matrix R .

Key words : Recommender System, Matrix Factorization, Collaborative Filtering, Accuracy, Social Network Analysis. Dimension Reductions
xviii+59 pages : 9 pictures; 12 tables, 1 algorithms
Bibliography : 45 (1992-2012)

NOTASI

Vektor dituliskan dengan huruf kecil yang bercetak tebal misalnya: \mathbf{w}_i , \mathbf{h}_j dan seterusnya.

Matriks ditulis dengan huruf kapital seperti : W , H dan seterusnya.

\mathbf{w}_i adalah dari vektor yang diambil dari baris ke i matriks W .

w_{ik} adalah elemen ke (i, k) dari matriks W

N : jumlah pengguna

M : jumlah produk

$i \in \{1, \dots, N\}$: indeks pengguna

$j \in \{1, \dots, M\}$: indeks produk

$r_{i,j}$: rating yang diberikan pengguna i pada produk j

$\hat{r}_{i,j}$: prediksi untuk rating $r_{i,j}$

R : matriks yang berukuran $n \times m$, dimana entrinya adalah $r_{i,j}$

D : Himpunan keseluruhan data rating

dimana $D = \{(q, (i, j, r_{ij})) | q \in Q, i \in U, j \in F, r_{ij} \in T\}$

$\mathcal{R} \subset D$:himpunan data training

$\mathcal{V} \subset D$:himpunan data testing

$\hat{e}_{i,j} = r_{i,j} - \hat{r}_{i,j}$: nilai error untuk pada elemen $(q, (i, j, r_{ij}))$

n_i : jumlah produk yang dirating oleh pengguna i , (jumlah entri pada baris ke- i) yang didefinisikan : $n_i = |\{j: (q, (i, j, r_{ij})) \in D\}|$.

n_j : jumlah pengguna yang merating produk j , (jumlah entri pada kolom ke- j) yang didefinisikan : $n_j = |\{i: (q, (i, j, r_{ij})) \in D\}|$.

k : banyaknya fitur yang dipilih dalam faktorisasi matriks

$W \in \mathbb{R}^{n \times k}$: matriks fitur pengguna

$H \in \mathbb{R}^{k \times m}$: matriks fitur produk

$\mathbf{w}_i \in \mathbb{R}^{k \times 1}$ vektor fitur pengguna, yaitu vektor yang elemen-elemennya adalah baris ke i matriks W

$\mathbf{h}_j \in \mathbb{R}^{k \times 1}$ vektor fitur produk yaitu, yaitu vektor yang elemen-elemennya adalah kolom ke j matriks H

w_{ik} : elemen ke (i, k) dari matriks W

h_{kj} : elemen ke (k, j) dari matriks H

I : matriks identitas

λ : Parameter regularisasi

η : *learning rate* yang dipakai pada metode *gradien descent*

\bar{y} : sampel mean dari sejumlah n pengamatan variabel acak y_1, y_2, \dots, y_n

σ^2 : varian populasi

σ_{xy} : kovarian populasi dari dua variabel acak X dan Y

Σ : Matriks varian kovarian populasi

$k \ll (n, m)$: nilai k jauh lebih besar dari nilai m dan n

pdf : *probability density function*

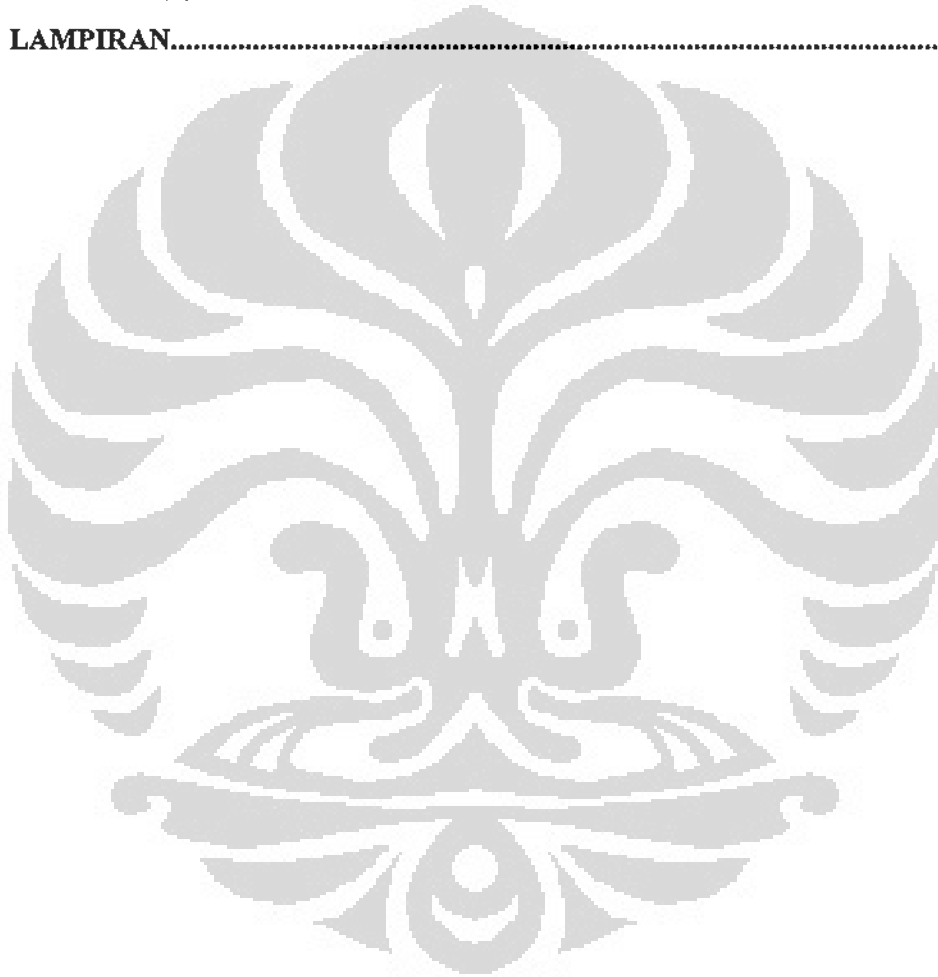
Variabel acak biasanya ditulis dalam huruf besar romawi: X, Y, dll. Realisasi dari sebuah variabel acak yang sesuai ditulis dengan huruf kecil.

Misalnya $x_1, x_2, x_3, \dots, x_4$ bisa menjadi sampel yang sesuai dengan variabel acak X.

DAFTAR ISI

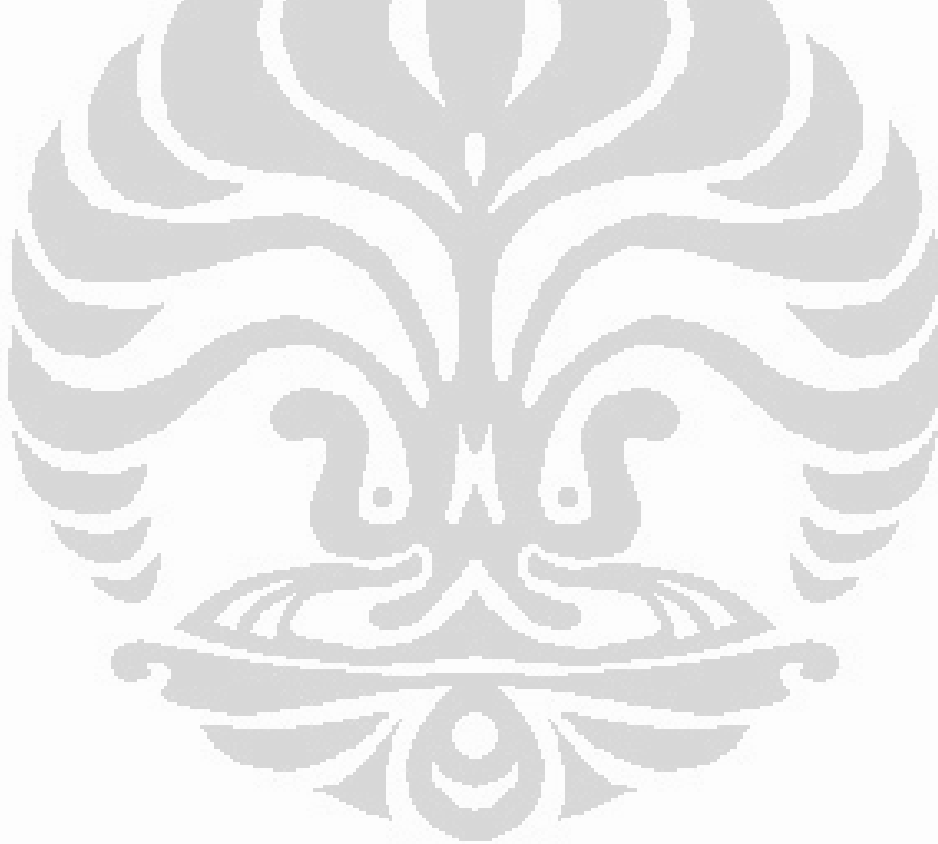
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN.....	iii
HALAMAN PERNYATAAN ORISINALITAS.....	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	viii
ABSTRAK.....	ix
ABSTRACT.....	x
NOTASI.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvi
DAFTAR ALGORITMA.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Permasalahan	6
1.3. Tujuan Penelitian.....	7
1.4. Metodologi Penelitian.....	7
BAB 2 LANDASAN TEORI	8
2.1. Sistem Rekomendasi.....	8
2.2. Model CF	10
2.3. Teori Matriks	12
2.4. Teori Probabilitas.....	16
2.5. Metode Bayesien	18
BAB 3 FAKTORISASI MATRIKS PROBABILITAS.....	21
3.1. Faktorisasi Matriks	21
3.2. Faktorisasi Matriks Probabilitas	22
3.3. Algoritma Faktorisasi Matriks Probabilitas.....	34
3.4. Ilustrasi dan Simulasi CF.....	36
BAB 4 SIMULASI.....	39
4.1. Data.....	39
4.2. Spesifikasi komputer	40
4.3. Desain Eksperimen.....	41
4.4. Inisialisasi Proses Simulasi.....	43
4.5. Akurasi Model CF	44

4.6.	Hasil eksperimen	45
4.6.1	Mencari nilai k	45
4.6.2	Mencari nilai λ	47
4.6.3	Menentukan pengaruh learning rate η terhadap nilai RMSE ...	49
4.7.	Matriks Prediksi	52
4.8.	Interpretasi hasil penelitian	52
BAB 5 PENUTUP		54
5.1.	Kesimpulan	54
5.2.	Saran	54
DAFTAR PUSTAKA		56
LAMPIRAN		60



DAFTAR GAMBAR

Gambar 1-1 Ilustrasi proses rekomendasi pada situs www.amazon.com	2
Gambar 2-1 Proses pembentukan prediksi dan rekomendasi. (Ercan, 2010)	12
Gambar 3-1 <i>Framework Machine Learning</i> untuk aplikasi Faktorisasi Matriks Probabilitas (Liu, 2011)	33
Gambar 3-2 Ilustrasi Matriks W dan H^T hasil eksperimen	38
Gambar 4-1 Skema percobaan model faktorisasi matriks	41
Gambar 4-2 Kurva RMSE untuk berbagai nilai k pada <i>fold</i> yang berbeda	47
Gambar 4-3 Kurva RMSE berdasarkan pemilihan nilai λ	48
Gambar 4-4 Kurva RMSE terhadap berbagai η	50
Gambar 4-5 Perbandingan RMSE Data Training dan Data Testing dalam setiap Iterasi	51

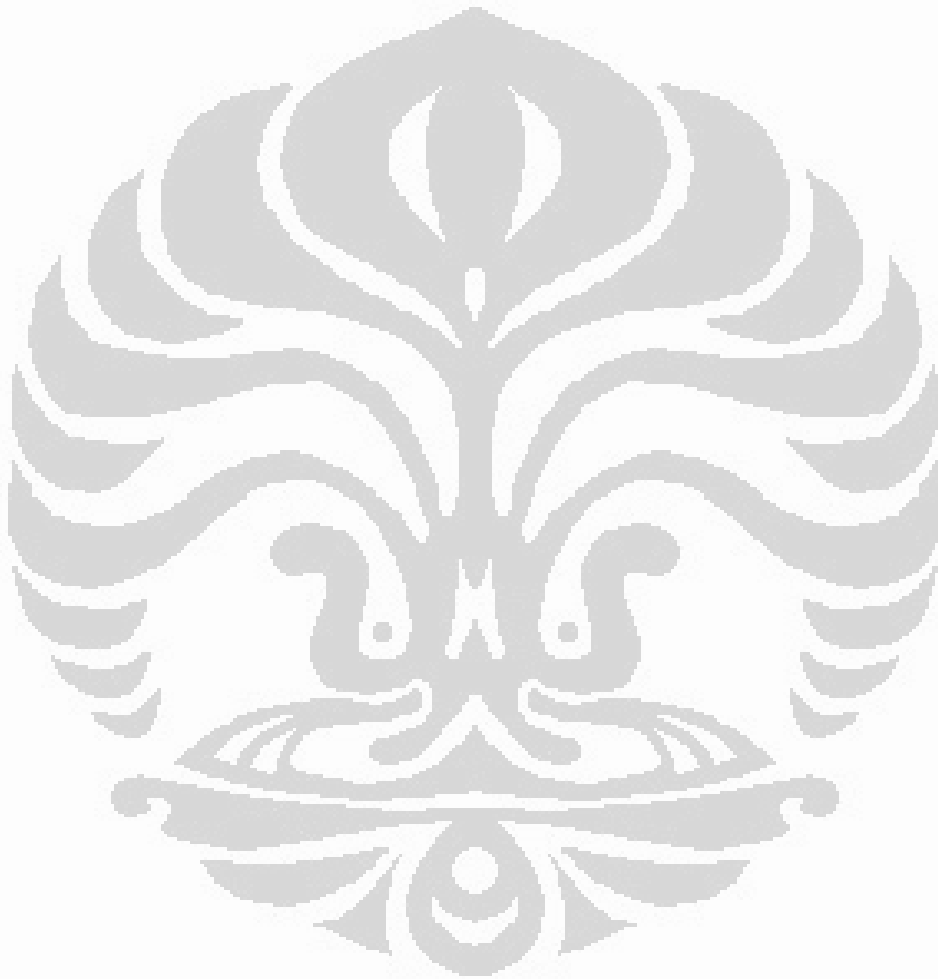


DAFTAR TABEL

Tabel 1-1 Data rating produk.....	5
Tabel 2-1 Ilustrasi data rating film (Graus, 2011).....	11
Tabel 2-2 Model Bayesien : <i>Posterior, Likelihood dan Prior</i> (Alpaydin, 2004).....	20
Tabel 3-1 Deskripsi data rating.....	36
Tabel 3-2 Format susunan data dalam komputer.....	37
Tabel 4-1 Indikator Data Penelitian.....	40
Tabel 4-2 Komposisi Data <i>Training</i> dan Data <i>Testing</i> untuk setiap <i>Fold</i>	40
Tabel 4-3 Desain eksperimen penentuan nilai parameter model.....	42
Tabel 4-4 RMSE untuk berbagi nilai k mulai dari <i>Fold 1</i> sampai dengan <i>Fold 5</i>	46
Tabel 4-5 Nilai RMSE terkecil dan kondisi parameter yang diperoleh dari Tabel 4-4.....	46
Tabel 4-6 Nilai RMSE berdasarkan perubahan nilai parameter regularisasi λ	48
Tabel 4-7 Nilai RMSE berdasarkan perubahan η	49

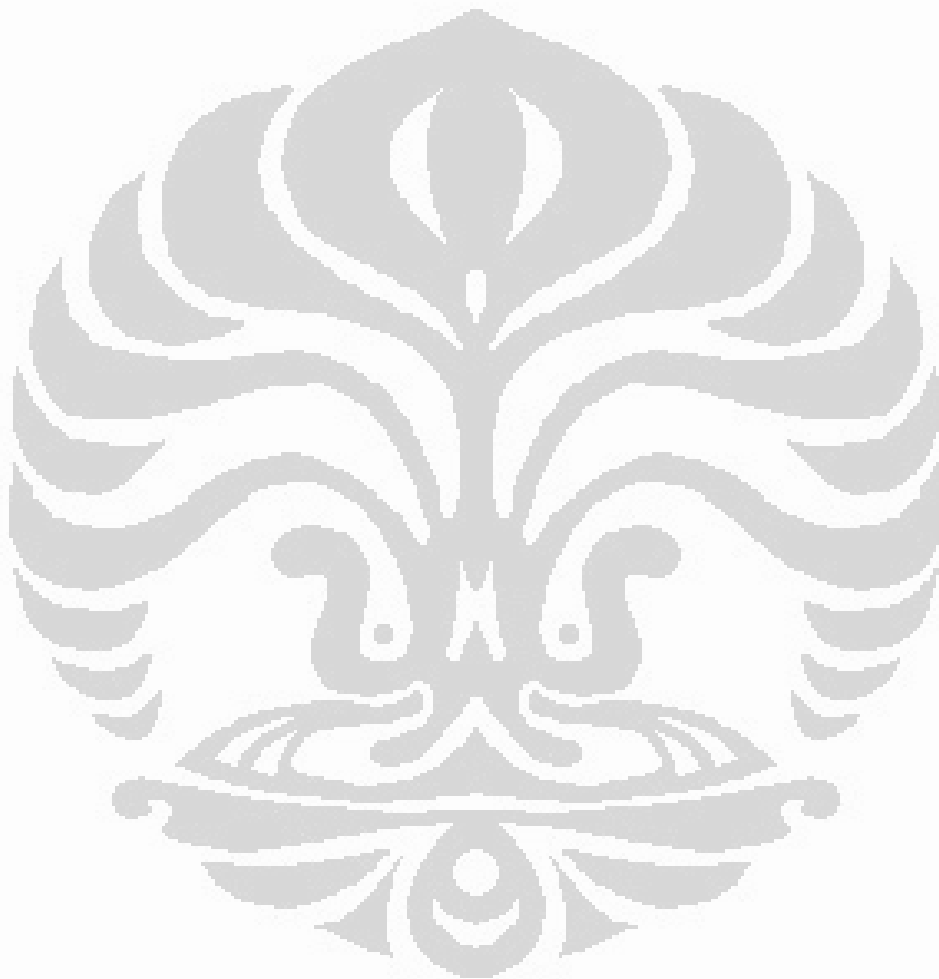
DAFTAR ALGORITMA

Algoritma 3-1: Algoritma Faktorisasi Matriks Probabilitas	35
---	----



DAFTAR LAMPIRAN

Lampiran 1 Ilustrasi Data rating	61
Lampiran 2 elemen himpunan \mathcal{R} dan \mathcal{V}	63



BAB 1

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi saat ini memiliki peranan yang begitu besar bagi peradaban manusia. Kemajuan ini telah merambah hampir di semua sektor kehidupan, terutama dunia industri yang merupakan salah satu roda penggerak perekonomian. Banyak industri telah memperoleh manfaat yang begitu berarti dari kemajuan teknologi.

Dengan diketemukannya teknologi internet, hampir semua bidang termasuk pertahanan, pendidikan, dan industri telah mengalami kemajuan yang sangat pesat. Kemajuan ini tak lepas dari banyaknya hasil riset unggulan yang dihasilkan universitas. Universitas dan industri harus senantiasa berkolaborasi untuk menciptakan produk riset unggulan. Oleh karena itu mahasiswa yang merupakan salah satu entitas di universitas diharapkan mampu menghasilkan karya ilmiah terbaik melalui penulisan thesis dan disertasi.

Salah satu hasil riset terkenal hasil karya universitas seperti ARPANET yang dikembangkan pada tahun 1969, telah mengilhami munculnya beberapa inovasi produk. Produk tersebut merupakan cikal bakal terbentuknya jaringan internet. Proyek ini telah diuji coba pada beberapa lembaga riset dan universitas ternama di Amerika Serikat. Dampak kemajuan teknologi tersebut sangat luar biasa bagi perkembangan teknologi di masa depan (Leiner, et al., 2009).

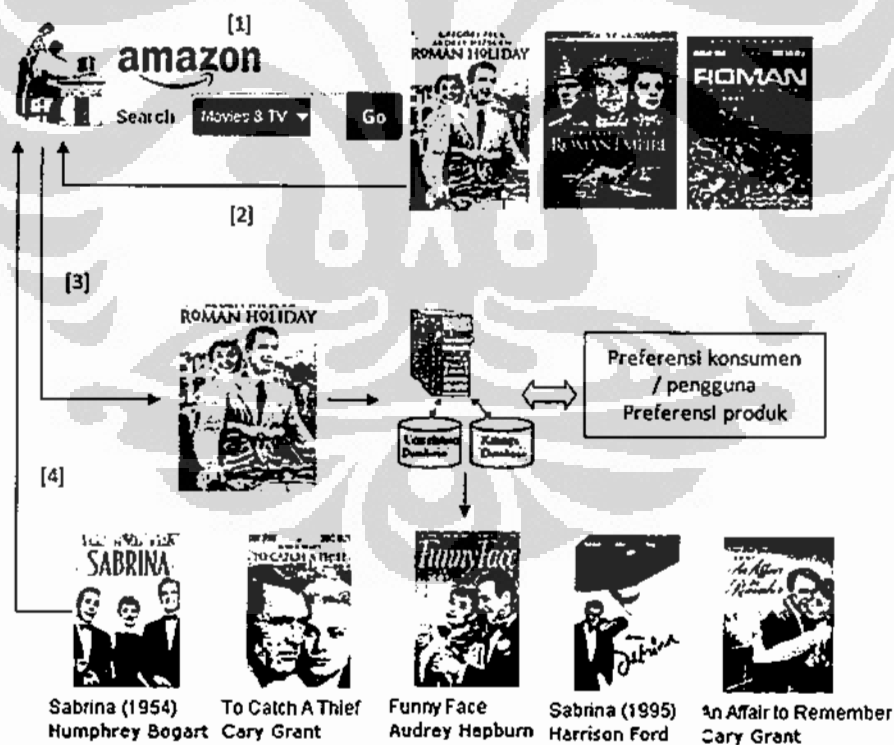
Salah satu dampak signifikan dari perkembangan teknologi internet telah terjadi dalam sistem penjualan. Perkembangan teknologi telah mengubah sistem penjualan dari pola tatap muka menjadi proses penjualan yang berbasis internet. Dengan teknologi internet, penjualan, pemasaran, manajemen mata rantai produk dari mulai pemasok, penjual dan pelanggan dapat dilakukan secara *online*. Beberapa perusahaan yang menjual dan memasarkan produknya secara *online* antara lain Amazon.com, MovieLens.org dan Netflix.com.

Walaupun banyak perusahaan telah memasarkan produknya secara *online*, namun belum semua perusahaan memiliki sistem penjualan yang mampu memprediksi minat pengguna atas semua produk yang ditawarkan. Untuk

mengatasi masalah ini diperlukan suatu teknik atau metode yang mampu memprediksi selera pengguna. Sistem inilah yang kemudian dikenal dengan sistem rekomendasi.

Terdapat beberapa definisi terkait dengan sistem rekomendasi diantaranya : Sarwar et al. (2001) mendefinisikan sistem rekomendasi sebagai suatu sistem yang dapat menyaring dan memberikan informasi yang sesuai dengan selera pengguna sebelumnya yang mempunyai selera yang sama. McGinty & Smyth (2006) mendefinisikan sistem rekomendasi sebagai sebuah model aplikasi dari sebuah observasi terhadap keadaan dan keinginan pengguna. Sistem rekomendasi memanfaatkan opini seseorang terhadap suatu barang dalam domain atau kategori tertentu untuk membantu seseorang dalam memilih dan menentukan suatu produk yang diminatinya.

Proses pencarian produk dapat diilustrasikan dengan Gambar 1.1. Pada gambar tersebut seorang pengguna (*user*) sedang berusaha mencari beberapa film



Gambar 1-1 Ilustrasi proses rekomendasi pada situs www.amazon.com

(item) yang disukainya pada situs *online* www.amazon.com. Dalam pencarian tersebut prosesnya dapat dijelaskan dengan informasi berikut:

1. Pengguna menghubungi situs www.amazon.com yaitu salah satu situs perusahaan yang menjual produknya secara *online*. Tampilan pertama yang muncul dari situs tersebut adalah halaman yang memuat sitem pencarian produk. Pada sistem pencarian tersebut, pengguna dipersilahkan mengetik kata kunci yang sesuai dengan produk yang diperlukan. Misalkan produk yang dicarinya adalah produk yang memiliki kata kunci '*Roman*'.
2. Dengan segera sistem merespon permintaan tersebut dengan menampilkan semua judul film yang memiliki kata '*Roman*'. Selanjutnya dari tampilan judul-judul film tersebut pengguna dapat menandai salah satu atau beberapa judul film yang memiliki kata '*Roman*' dari film-film yang ditampilkan. Misalkan salah satu film yang ditandai oleh pengguna adalah film '*Roman Holiday*'.
3. Selanjutnya sistem mencatat pilihan pengguna tersebut dan kemudian membandingkan antara data input yang diberikan pengguna dengan beberapa informasi yang ada di basis data diantaranya : preferensi pengguna, preferensi produk, dan catatan penjualan yang terkait dengan produk film '*Roman Holiday*'.
4. Terakhir, sistem merekomendasikan pada pengguna beberapa produk film yang memiliki kedekatan dengan film '*Roman Holiday*', yaitu: "*Sabrina* ", "*To Catch a Tief*", "*Funny Face*", dan "*An affair to remember*".

Keempat tahapan tersebut merupakan ilustrasi sederhana yang dapat menjelaskan proses yang bekerja pada sistem rekomendasi untuk penjualan produk yang dilakukan secara *online*.

Salah satu metode yang telah dipakai untuk menjelaskan ilustrasi seperti yang terlihat pada Gambar 1.1 adalah CF, yaitu suatu teknik membuat prediksi serta rekomendasi tentang minat seorang pengguna berdasarkan pengumpulan informasi atau opini dari banyak pengguna sebelumnya (Sarwar et al., 2001).

Selain memprediksi, sistem rekomendasi dapat membantu pengguna dalam mencari produk-produk yang disukainya, sebaliknya dari sisi penjual sistem ini diharapkan mampu meningkatkan jumlah penjualan (Sarwar et al., 2000). Kemampuan sistem dalam memprediksi dan merekomendasikan suatu produk akan sangat bermanfaat baik bagi pengguna maupun bagi penjual.

Oleh karena itu sistem penjualan yang dibangun harus memiliki kemampuan memprediksi dan merekomendasi. Sejalan dengan hal tersebut, menurut Bishop (2006) sistem rekomendasi perlu dilengkapi dengan aplikasi cerdas yang mampu melakukan generalisasi terhadap data baru yang belum dipelajari sebelumnya dalam hal mengklasifikasi dan memprediksi.

Kemampuan sistem penjualan dalam memprediksi dan merekomendasi dapat dipelajari melalui teknik atau metode yang terdapat dalam *machine learning*. Bishop (2006) menyebutkan ada dua jenis model pembelajaran yaitu *supervised learning* dan *unsupervised learning*. Dalam sistem rekomendasi, data *training* yang dianalisa tidak disertai target pembelajaran. Tujuan pembelajarannya adalah membangun model yang dapat menemukan variabel atau komponen tersembunyi pada data *training*, sehingga model yang terbentuk dapat digunakan untuk membuat prediksi dan rekomendasi. Oleh karena itu *learning model* yang cocok diterapkan dalam sistem rekomendasi adalah *unsupervised learning model*.

Merujuk pada Hao (2009) sistem rekomendasi dapat menggunakan salah satu dari dua strategi berikut ini, yaitu *content filtering* dan *CF*. Dengan strategi *content filtering*, baik untuk pengguna maupun produk profilnya harus didefinisikan terlebih dahulu di awal penelitian.

Untuk produk film, profil tersebut berupa genre, aktor, popularitas, dan lain-lain. Sedangkan profil pengguna dapat berupa informasi geografis dan lain-lain. Profil-profil ini memungkinkan sistem dapat mengasosiasikan pengguna dengan produk-produk yang sesuai. Strategi berbasis konten ini membutuhkan informasi eksternal yang boleh jadi sangat sulit untuk diperoleh. Sedangkan dalam strategi *CF*, rekomendasi hanya bergantung pada aktifitas pengguna sebelumnya, misalnya data pada transaksi atau pemberian rating, tanpa mendefinisikan profil secara eksplisit.

Dalam *CF* sendiri ada dua pendekatan yaitu: *nearest neighbors* dan *latent variable models*. Penelitian ini hanya membatasi pada pendekatan *latent variable model*. Dalam hal ini prediksi rating yang dibangun didasarkan pada posisi relatif dari pengguna dan produk pada beberapa variabel tersembunyi yang diekstrak dari pola rating. Untuk film, variabel tersembunyi tersebut berupa *genre*, misalnya : Petualangan, Animasi, Komedi, Kriminal, Dokumenter, Drama, Fantasi, Horor, Misteri, Roman dan lainnya. Salah satu metode yang mampu menangani masalah *latent variable models* adalah *matrix factorization*. Metode ini menjadi populer karena mampu memberikan skalabilitas yang baik dengan akurasi yang prediktif.

Dengan metode ini biasanya data *rating* direpresentasikan dalam bentuk matriks, dimana baris-baris pada matriks menyatakan pengguna, sementara kolom-kolomnya menyatakan produk. Matriks tersebut biasa ditulis dengan $R_{n \times m}$, dimana entri r_{ij} dari matriks $R_{n \times m}$, menyatakan rating produk yang diberikan oleh pengguna i pada produk j . Bila digambarkan, matriks tersebut dapat diilustrasikan dengan Tabel 1-1.

Tabel 1-1: Data rating produk

	Produk ke-1	Produk 2	...	Produk m
Pengguna 1	r_{11}	r_{12}	...	r_{1m}
Pengguna 2	r_{21}	r_{22}	...	r_{2m}
...
Pengguna n	r_{n1}	r_{n2}	...	r_{nm}

Pada umumnya jumlah entri rating pada matriks tersebut jauh lebih kecil dibandingkan dengan banyaknya entri maksimum dalam matriks yang berukuran $n \times m$, sehingga matriks yang diperoleh dari data rating tersebut akan membentuk matriks *sparse*. Hal ini bisa dipahami, karena seorang pengguna umumnya hanya memberikan rating pada sebagian kecil produk saja. Oleh karena itu penggunaan *singular value decomposition* (SVD) untuk mengekstrak variabel tersembunyi seperti pada *latent semantic analysis* (LSA) hanya bisa dilakukan dengan sejumlah cara. Dengan cara tersebut, entri yang tidak diketahui dapat diganti dengan nol (*imputation*). Namun cara ini dapat menyebabkan peningkatan jumlah data yang tidak dilakukan secara benar atau merusak data.

Sarwar et al., (2001) menyebutkan bahwa algoritma sistem rekomendasi yang menggunakan *nearest neighbor* tidak mungkin dibuat untuk matrik dengan *sparsity* yang besar, bila hal ini dipaksakan maka kualitas rekomendasi yang dihasilkan menjadi sangat rendah.

Untuk itu diperlukan suatu pendekatan yang dapat merumuskan pemodelan yang bisa diperoleh dari data rating yang diketahui saja (*data training*). Salah satu formulasi yang bisa digunakan untuk menyelesaikan masalah tersebut adalah dengan menerapkan metode faktorisasi matriks.

Dalam faktorisasi matriks, jika diberikan suatu matriks $R_{n \times m}$, dimana n adalah banyaknya pengguna, m menyatakan banyaknya produk sedangkan k adalah banyaknya variabel laten yang terdapat pada produk maupun pengguna, maka masalah faktorisasi matriks adalah mencari matriks $W_{n \times k}$ dan $H_{k \times m}$ sedemikian sehingga $WH \approx R$. Nilai k umumnya jauh lebih kecil dari n dan m . Semakin kecil selisih WH dan R , maka model yang terbentuk akan semakin baik.

Terdapat beberapa artikel yang telah dipublikasikan di berbagai jurnal membahas tentang faktorisasi matriks antara lain : *Algorithms for Non-Negative Matrix Factorization* (Lee & Seung, 2000); *Fast Maximum Margin Matrix Factorization* (Rennie & Srebro, 2005); *List-wise Learning to Rank with Matrix Factorization* (Shi, Larson, & Hanjalic, 2010); dan *Probabilistic Matrix Factorization* (Salakhutdinov & Mnih, 2008). Dalam tesis ini metode faktorisasi matriks yang digunakan adalah faktorisasi matriks probabilitas. Penjelasan mengenai metode ini diuraikan lebih lengkap dalam Bab 3, yaitu bab yang membahas faktorisasi matriks.

1.2. Permasalahan

Berdasarkan uraian yang disampaikan dalam subbab 1.1, maka permasalahan yang diajukan dalam tesis ini adalah:

- Bagaimana menentukan parameter k yang optimal dari metode aproksimasi faktorisasi matriks probabilitas untuk mendapatkan matriks W dan H yang optimal ?

- Bagaimana mengukur keakuratan metode aproksimasi faktorisasi matriks probabilitas terhadap data real?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah mencari solusi atau metode yang dapat menjawab beberapa hal yang telah diuraikan dalam subbab permasalahan di atas yaitu :

- Menentukan parameter model yang optimal dari faktorisasi matriks probabilitas.
- Menghitung akurasi dari metoda faktorisasi matriks probabilitas terhadap data real.

1.4. Metodologi Penelitian

Penelitian dilakukan berdasarkan langkah-langkah berikut :

- **Studi Literatur**
Mempelajari berbagai sumber bacaan baik dari jurnal, studi kasus maupun disertasi yang membahas berbagai penelitian mengenai Sistem Rekomendasi, *CF*, Faktorisasi Matriks dan Teori Probabilitas.
- **Pembuatan Model**
Berdasarkan informasi yang diperoleh dari studi literatur, langkah yang dilakukan selanjutnya adalah membuat pemodelan dan memprogramnya dengan Matlab
- **Pengujian Program**
Menerapkan data MovieLens (movielens.org) ke dalam program untuk dianalisa lebih lanjut. Data ini diperoleh dari situs www.movielens.org yang dikelola oleh *GroupLens Research Project* Universitas Minnesota. Semuanya terdiri dari : 100,000 data rating dengan skala (1 sd 5) dari 943 pengguna pada 1682 produk.

BAB 2

LANDASAN TEORI

Beberapa tinjauan teoritis yang mendasari teori faktorisasi matriks probabilitas untuk sistem rekomendasi diuraikan dalam bab ini. Bab ini terbagi ke dalam lima subbab yaitu : subbab pertama membahas sistem rekomendasi, subbab kedua membahas model *CF*, subbab ketiga membahas teori matriks, subbab keempat membahas teori probabilitas, dan subbab kelima membahas metode Bayesian.

2.1. Sistem Rekomendasi

Hao (2009) membagi sistem rekomendasi ke dalam dua kategori yaitu:

- *Content-based filtering*
- *CF*

Di antara kedua kategori tersebut, pendekatan yang-dibahas dalam tesis ini adalah sistem rekomendasi yang menggunakan teknik *CF*. Metode *CF* merupakan metode yang banyak digunakan dalam sistem rekomendasi. Penggunaannya telah menyebar begitu luas baik di kalangan akademisi maupun kalangan industri (Hao, 2009). Sedangkan *Content-based filtering* tidak begitu banyak dipakai dalam sistem rekomendasi. Hal ini disebabkan adanya kesulitan menerapkan *Content-based filtering* yang mensyaratkan perlunya mendefinisikan profil pengguna maupun produk di awal penelitian.

Ada dua pendekatan yang diterapkan dalam *CF* yaitu pendekatan berbasis memori dan pendekatan berbasis model. Pendekatan berbasis memori merupakan suatu metode prediksi yang sudah terkenal dan telah begitu luas diterapkan pada bidang komersial seperti pada perusahaan amazon.com (Linden, Smith, & York, 2003).

Pendekatan *CF* berbasis memori telah banyak dijumpai di beberapa artikel di antaranya: *Item-Based Top-N Recommendation Algorithms* (Deshpande & Karypis, 2004); *Improving the Diversity of User-based Top-N Recommendation by Cloud Model* (Wang, Tao, & Hu, 2010).

Strunjas (2008) mengategorikan pendekatan berbasis memori kedalam dua kategori, yaitu : metode berbasis pengguna dan metode berbasis produk. Metode berbasis pengguna adalah suatu teknik yang digunakan untuk menghitung kemiripan pengguna, sedangkan metode berbasis produk digunakan untuk menghitung kemiripan produk. Perumusan yang dipakai kedua metode tersebut adalah korelasi dan kosinus. Seberapa mirip pengguna yang satu dengan pengguna lainnya dapat dilihat dari nilai korelasinya. Disamping metode tersebut, terdapat metode yang menggabungkan metode berbasis pengguna dan metode berbasis model (Wang, de Vries, & Reinders, 2006).

Hao (2009) menyebutkan bahwa dalam *CF* berbasis pengguna, kemiripan antara dua pengguna dapat dihitung dengan menggunakan *Pearson Correlation Coefficient* sebagai berikut:

$$\text{sim}(a, b) = \frac{\sum_{j \in J(a) \cap J(b)} ((r_{a,j} - \bar{r}_a)(r_{b,j} - \bar{r}_b))}{\sqrt{\sum_{j \in J(a) \cap J(b)} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{j \in J(a) \cap J(b)} (r_{b,j} - \bar{r}_b)^2}} \quad (2.1)$$

dimana :

- $\text{Sim}(a,b)$: menyatakan kemiripan antara pengguna a dan pengguna b
- $J(a)$: semua produk yang dirating oleh pengguna a
- $J(b)$: semua produk yang dirating oleh pengguna b
- $J(a) \cap J(b)$: produk-produk sama-sama dirating baik oleh pengguna a maupun oleh pengguna b
- $r_{a,j}$: rating yang diberikan oleh pengguna a pada produk j
- $r_{b,j}$: rating yang diberikan oleh pengguna b pada produk j
- \bar{r}_a : rata-rata rating pengguna a
- \bar{r}_b : rata-rata rating pengguna b

Pembahasan metode *CF* berbasis pengguna maupun berbasis produk, sudah banyak ditulis dalam beberapa karya ilmiah seperti: *User-based Collaborative Filtering Recommendation Algorithms on Hadoop* yang dipublikasikan oleh Zhao & Shang (2010) dan *Item-based collaborative filtering*

recommender system algorithms oleh Sarwar et al. (2001) untuk metode CF berbasis produk.

Berikut adalah beberapa artikel yang membahas CF berbasis model diantaranya : *Bayesian Network* (Su, 2008); *Latent Semantic Model* (Hofmann, 2004); *MDP-Markov decision process* (Shani, Heckerman, & Brafman, 2005); *SVD* (Zhang, Wang, Ford, & Makedon, 2005); dan *Probabilistic Matrix Factorization* (Salakhutdinov & Mnih, 2008).

Salah satu kelebihan yang bisa diperoleh dengan menerapkan metode CF berbasis model adalah kemampuannya dalam menangani *sparsity* dan *scalability* data yang lebih baik dibandingkan dengan metode CF yang berbasis memori (Gong, Ye, & Tan, 2009). Pemrosesan data dalam skala besar akan lebih mudah bila menerapkan CF berbasis model. Dengan metode ini, kinerja model yang dihasilkan akan jauh lebih baik dibandingkan dengan metode CF berbasis memori. Hal inilah yang menjadi alasan mengapa metode CF berbasis model lebih baik diterapkan dalam sistem rekomendasi dibandingkan dengan metode CF berbasis memori (Su & Khoshgoftaar, 2009).

Oleh karena itu dalam penelitian ini penulis menerapkan metode CF berbasis model faktorisasi matriks. Pemilihan metode ini didasarkan pada *sparsity* dan *scalability* pada data penelitian yang digunakan. Uraian lebih jauh mengenai metode ini akan dibahas lebih lengkap pada bab tiga yaitu faktorisasi matriks.

2.2. Model CF

Pilaszy (2009) dalam disertasinya menyebutkan bahwa dalam CF, hubungan pengguna, produk dan rating dapat dinyatakan dalam bentuk triplel (I,J,Y) dimana:

$I = \{1, \dots, N\}$, I menyatakan himpunan indeks pengguna, N jumlah pengguna.

$J = \{1, \dots, M\}$, J menyatakan himpunan indeks produk, M jumlah produk.

$Y = \{1,2,3,4,5\}$, Y adalah himpunan yang elemennyamerupakan *rating* produk.

Produk yang tercantum dalam penelitian ini dinyatakan dalam judul film.

Sehingga bila seorang pengguna i merating film j dengan skor rating y maka hal tersebut dapat ditulis dengan (i, j, y) .

Graus (2011) menyebutkan bahwa sumber data yang di gunakan dalam *CF* dapat diperoleh dari data *rating*. Data tersebut umumnya ditampilkan dalam bentuk matriks, dimana baris-barisnya menyatakan indeks pengguna, sedangkan kolom-kolomnya menyatakan indeks film. Entri $r_{i,j}$ adalah nilai *rating* yang diberikan oleh pengguna i pada film j . *Rating* film menggunakan skala *likert* dengan rentang skor mulai dari 1 sd 5, dimana nilai 1: film sangat buruk, 2: film buruk, 3: film biasa saja, 4: film baik, dan rating 5 film sangat baik. Sedangkan untuk entri matriks $r_{i,j}$ yang kosong bukan berarti *rating* film yang diberikan pengguna i pada film j memiliki skor *rating* nol, akan tetapi kondisi tersebut hanya menunjukkan bahwa pengguna i tidak merating film j . Tabel 2-1 berikut adalah ilustrasi data skor *rating* yang dipilih oleh 4 orang pengguna pada empat judul film.

Pada Tabel 2-1 ditunjukkan bahwa tidak semua pengguna memberikan *rating* pada keempat film tersebut. Dengan kata lain ada beberapa judul film yang tidak diberikan *rating* oleh pengguna. Akibatnya ada entri pada baris dan kolom tertentu terlihat kosong. Kekosongan ini tentu saja dapat menimbulkan suatu pertanyaan. Sebagai contoh berapa skor *rating* yang diberikan oleh Mark pada film *Die Hard*?. Pertanyaan serupa juga disampaikan oleh Wen (2008) berikut ini:

$$\text{rating} = \begin{cases} r_{i,j} & : \text{rating yang diberikan oleh pengguna } i \text{ pada film } j \\ ? & : \text{Pengguna tidak memberikan } \textit{rating} \end{cases}$$

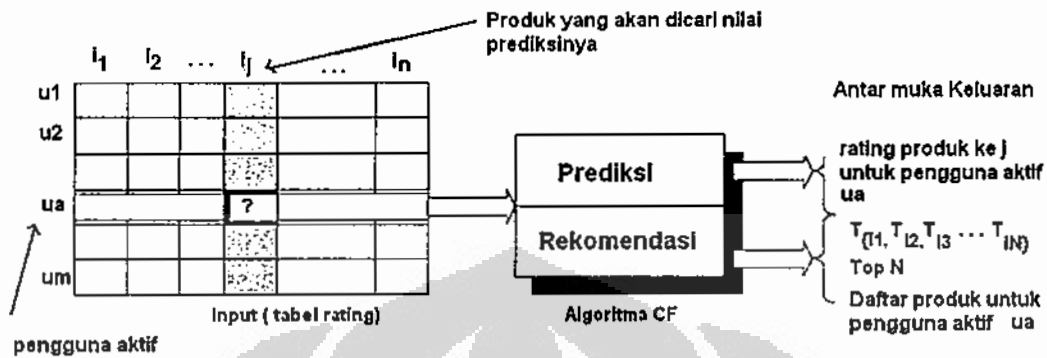
Tabel 2-1 Ilustrasi data *rating* film
(Graus, 2011)

	f_1	f_2	f_3	f_4
Jack		1	5	
Dylan			4	5
Olivia	2	4	1	2
Mark	1		?	

f_1 : *Usual Suspects*, f_2 : *Titanics*, f_3 : *Die Hard*, f_4 : *The Godfather*

Terdapat beberapa metode yang bisa diterapkan untuk menghitung prediksi skor *rating* seperti dalam kasus skor *rating* Mark pada film *Die Hard*.

Salah satu metode yang dikenal dalam *machine learning* adalah metode *CF*, yaitu suatu metode yang dapat diilustrasikan dengan Gambar 2-1.



Gambar 2-1 Proses pembentukan prediksi dan rekomendasi.

(Ercan, 2010)

Dalam *CF*, terdapat banyak metode, salah satunya adalah metode faktorisasi matriks. Seberapa baik metode faktorisasi matriks diterapkan dalam membangun model, indikatornya dapat dilihat dari besarnya RMSE yang dicapai oleh model tersebut pada data *testing*. Dalam hal ini RMSE (*root mean squared error*) adalah bilangan real yang nilainya lebih besar dari nol. Semakin kecil nilai RMSE, maka model yang diperoleh akan semakin baik. Informasi mengenai RMSE secara detil dapat dilihat di bab empat.

Berikut adalah landasan teori yang digunakan dalam faktorisasi matriks. Landasan teori tersebut umumnya disajikan dalam bentuk definisi dan teorema. Teorema yang dipakai sebagian besar bersumber dari teori matriks dan teori probabilitas.

2.3. Teori Matriks

Teori matriks merupakan landasan teori yang sebagian besar digunakan dalam faktorisasi matriks. Terdapat beberapa definisi dan teorema yang terkait dengan teori matriks yaitu : matriks invers, matriks transpos, determinan matriks, norm Frobenius dan operasi perkalian matriks Hadamard.

Definisi dan teorema matriks yang disampaikan pada subbab ini diambil dari berbagai sumber di antaranya dari : Roman (2008), Saad (2003), Liu (2011),

Caro-Lopera, Leiva, & Balakrishnan (2012), dan Rencher (2002). Berikut adalah beberapa definisi dan teorema mengenai matriks yang sebagian besar akan digunakan di bab tiga.

Definisi 2.1 (Liu, 2011)

Suatu matriks identitas I yang berukuran $n \times n$ adalah matriks persegi yang elemen-elemen diagonal utamanya sama dengan 1 sedang elemen yang lainnya sama dengan 0 atau ditulis,

$$i_{x,y} = \begin{cases} 1, & \text{jika } x = y \\ 0, & \text{jika } x \neq y \end{cases}$$

dimana $i_{x,y}$ adalah entri dari matriks identitas I .

Jika diberikan matriks $A_{n \times n}$ maka berlaku : $AI = A = IA$.

Definisi 2.2 (Liu, 2011)

Jika A adalah suatu matriks persegi $A \in \mathbb{R}^{n \times n}$, maka *trace* A dinyatakan dengan :

$$\text{tr}(A) = \sum_{i=1}^n a_{ii}$$

Misalkan $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ maka : $\text{tr}(A) = a_{11} + a_{22} + a_{33}$

Definisi 2.3 (Liu, 2011)

Suatu matriks persegi $A \in \mathbb{R}^{n \times n}$ adalah matriks yang dapat dibalik atau non singular jika ada matriks persegi $B \in \mathbb{R}^{n \times n}$ maka:

$$AB = I = BA$$

Dalam kasus ini, B secara unik ditentukan oleh A , maka B adalah invers dari matriks A atau ditulis A^{-1} , sedangkan bila B atau A^{-1} tidak ada, maka matriks A adalah matriks yang tidak dapat dibalik dan disebut matriks singular.

Definisi 2.4 (Roman, 2008)

Transpos dari matriks $A \in \mathbb{R}^{m \times n}$ adalah matriks A^T yang didefinisikan oleh:

$$(A^T)_{i,j} = (A)_{j,i}$$

Matriks A dikatakan simetrik jika $A = A^T$ dan skew simetrik jika $A^T = -A$

Definisi 2.5 (Roman, 2008)

Suatu matriks persegi disebut segitiga atas jika semua entri di bawah diagonal utamanya bernilai 0. Hal yang sama, suatu matriks persegi disebut segitiga bawah jika semua entri diatas diagonal utamanya bernilai 0. Matriks persegi disebut matriks diagonal jika semua entrinya selain diagonal utama bernilai 0.

Definisi 2.6 (Roman, 2008)

Misalkan V adalah ruang vektor atas lapangan F bilangan riil ($F = \mathbb{R}$) maka hasilkali dalam pada V adalah suatu fungsi $\langle \cdot, \cdot \rangle : V \times V \rightarrow F$ dengan sifat:

(Definit Positif) Untuk setiap $v \in V$, $\langle v, v \rangle \geq 0$ dan $\langle v, v \rangle = 0 \Leftrightarrow v = 0$

(Simetri) $\langle u, v \rangle = \langle v, u \rangle$

(Linieritas) untuk setiap $u, v, w \in V$ dan $r, s \in F$ maka

$$\langle ru + sv, w \rangle = r \langle u, w \rangle + s \langle v, w \rangle$$

Contoh :

Misalkan $u = (u_1, u_2, u_3, u_4)$; $v = (v_1, v_2, v_3, v_4)$ dan $w = (w_1, w_2, w_3, w_4)$ maka:

$$\langle u, v \rangle = u_1v_1 + u_2v_2 + u_3v_3 + u_4v_4 = \langle v, u \rangle$$

Teorema 2.7 (Roman, 2008)

Misalkan $A \in \mathbb{R}^{n \times n}$ dan $\forall B \in \mathbb{R}^{n \times n}$, maka :

$$\det(AB) = \det(A) \det(B)$$

A adalah non singular (dapat dibalik) jika dan hanya jika $\det(A) \neq 0$

Determinan dari matriks segitiga atas, atau matriks segitiga bawah sama dengan perkalian entri-entri pada diagonal utamanya.

Contoh:

$$\text{Misalkan } \Sigma = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

Maka :

$$\det(\Sigma) = |\Sigma| = a_{11}a_{22}a_{33}$$

Definisi 2.8 (Saad, 2003)

Jika $A = (a_{i,j})$ adalah matriks ukuran $m \times n$, maka norm Frobenius dari matriks A adalah :

$$\|A\|_F = \left(\sum_{i,j} a_{i,j}^2 \right)^{1/2}$$

Contoh :

$$\text{Misalkan } A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \text{ maka :}$$

$$\|A\|_F = (a_{11}^2 + a_{12}^2 + a_{13}^2 + a_{21}^2 + a_{22}^2 + a_{23}^2 + a_{31}^2 + a_{32}^2 + a_{33}^2)^{1/2}$$

Definisi 2.9 (Caro-Lopera, Leiva, & Balakrishnan, 2012)

Misalkan $A, B \in \mathbb{R}^{m \times n}$ adalah dua matriks yang berukuran sama dan entri matriks ke (i,j) dari matriks A dan B masing-masing adalah $a_{i,j}$ dan $b_{i,j}$ maka Perkalian Hadamard matriks A dengan matriks B ditulis $A \odot B$ yang didefinisikan:

$$(A \odot B)_{i,j} = a_{i,j}b_{i,j}$$

Misalkan matriks A dan B adalah

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \text{ dan } B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

maka perkalian Hadamard matriks A dengan matriks B adalah

$$A \odot B = \begin{bmatrix} a_{11} * b_{11} & a_{12} * b_{12} & a_{13} * b_{13} \\ a_{21} * b_{21} & a_{22} * b_{22} & a_{23} * b_{23} \\ a_{31} * b_{31} & a_{32} * b_{32} & a_{33} * b_{33} \end{bmatrix}$$

Definisi 2.10 (Liu, 2011)

Jika diberikan dua vektor $x, y \in \mathbb{R}^n$, maka kedua vektor tersebut adalah ortogonal jika : $x^T y = 0$.

Suatu matriks persegi $A \in \mathbb{R}^{n \times n}$ adalah matriks ortogonal jika setiap dua vektor kolomnya merupakan ortogonal.

$$A^T A = I = A A^T$$

Definisi 2.11 (Liu, 2011)

Jika diberikan vektor $x \in \mathbb{R}^n$ dan sebuah matriks persegi $A \in \mathbb{R}^{n \times n}$, maka bentuk kuadrat skalarnya adalah:

$$x^T A x = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

2.4. Teori Probabilitas

Teori probabilitas merupakan salah satu cabang matematika yang banyak digunakan hampir di setiap cabang ilmu pengetahuan, seperti biologi, astronomi, fisika, kedokteran, ekonomi, kimia, sosiologi, ekologi, keuangan, dan sebagainya (DasGupta, 2011). Dalam metode faktorisasi matriks probabilitas, teori probabilitas memiliki peranan yang sangat penting dalam membangun sistem rekomendasi. Model faktorisasi ini sudah dirintis oleh Salakhutdinov & Mnih pada tahun 2008.

Terdapat beberapa istilah yang sering dipakai dalam teori probabilitas di antaranya adalah :

- Ruang sampel Ω adalah himpunan yang elemen-elemennya merupakan semua hasil yang mungkin dari suatu eksperimen. Himpunan ini menjadi dasar proses yang akan dilakukan selanjutnya.
- Titik sampel adalah elemen dari ruang sampel
- Peristiwa (*event*) A adalah himpunan bagian dari ruang sampel Ω
- Percobaan acak adalah suatu eksperimen yang dapat diulang pada kondisi yang sama dimana hasil dari percobaan tersebut tidak dapat ditentukan dengan pasti, tetapi koleksi dari semua hasil yang mungkin dapat diketahui.

Berikut adalah beberapa definisi dan teorema probabilitas yang digunakan dalam faktorisasi matriks probabilitas:

Definisi 2.12. (DasGupta, 2011)

Dalam ruang probabilitas (Ω, F, P) dimana Ω adalah ruang sampel, F ruang kejadian dan P adalah probabilitas maka berdasarkan Aksioma Kolmogorov:

- (a) $P(A) \geq 0$ untuk setiap $A \in F$,
- (b) $P(\Omega) = 1$;
- (c) subhimpunan yang saling lepas untuk kejadian A_1, A_2, \dots, A_n membentuk partisi terhadap Ω : $A_i \cap A_j = \emptyset, \forall i \neq j$, sehingga

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i)$$

Proposisi 2.13. (DasGupta, 2011)

Misalkan Ω adalah ruang sampel berhingga yang terdiri dari N titik sampel yang memiliki peluang yang sama untuk terjadi. Misalkan A adalah kejadian dan anggap A mengandung n titik sampel yang berbeda, maka:

$$P(A) = \frac{n}{N} = \frac{\text{jumlah titik sampel pada kejadian } A}{\text{Jumlah keseluruhan titik sampel}}$$

Definisi 2.14. (DasGupta, 2011)

Misalkan A, B adalah kejadian-kejadian pada ruang sampel Ω , dan misalkan $P(A) > 0$. Probabilitas bersyarat B jika diberikan A didefinisikan sebagai

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Teorema 2.15. (DasGupta, 2011)

Misalkan $\{A_1, A_2, \dots, A_m\}$ adalah partisi dari ruang sampel Ω . $A_i \cap A_j = \emptyset, \forall i \neq j$ $\bigcup_{i=1}^m A_i = \Omega$ dan suatu kejadian lain B , (Probabilitas $P(A_j|B)$ adalah probabilitas posteriori dari A_j)

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^m P(B|A_i)P(A_i)}$$

Definisi 2.16. (DasGupta, 2011)

Misalkan X dan Y adalah dua variabel acak yang didefinisikan dalam ruang sampel Ω . Maka kovarian dari X dan Y didefinisikan sebagai

$$\text{Cov}(X, Y) = \sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)] = E(XY) - \mu_X\mu_Y$$

Jika X dan Y saling bebas, maka $E(XY) = E(X)E(Y)$, sehingga $\sigma_{xy} = 0$

Definisi 2.17. (DasGupta, 2011)

Variabel acak X dengan parameter *mean* μ dan varian σ^2 dikatakan berdistribusi normal atau ditulis $X \sim \mathcal{N}(\mu, \sigma^2)$ jika fungsi *densitas* variabel acak X memenuhi fungsi berikut:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}; -\infty < x < \infty,$$

Definisi 2.18 (Koch, 2007)

Misalkan $X = (X_1, X_2, \dots, X_d)$ adalah vektor dari variabel acak yang berdimensi d dan berdistribusi normal atau dilambangkan dengan $X \sim \mathcal{N}(\mu, \Sigma)$ dengan parameter mean $\mu = (\mu_1, \mu_2, \dots, \mu_d)^T$ yang merupakan vektor berukuran $d \times 1$ dan matriks kovarian Σ definit positif (non singular) yang berukuran $d \times d$. Maka variabel acak X dikatakan berdistribusi multivariat normal berdimensi d jika fungsi kepadatan probabilitas dari variabel acak X memenuhi fungsi berikut

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

2.5. Metode Bayesian

Salah satu pendekatan *CF* yang paling populer dalam faktorisasi matriks adalah mentransformasi variabel data *rating* ke dalam variabel tersembunyi. Model yang terbentuk didasarkan pada suatu kenyataan bahwa sikap atau *preferensi* pengguna dapat ditentukan oleh sejumlah kecil variabel baru yang disebut dengan variabel laten

Dalam faktorisasi matriks, variabel ini dapat diwujudkan dalam dua buah matriks faktorisasi. Wujud kedua matriks faktorisasi tersebut bisa diperoleh

melalui suatu pendekatan atau taksiran (*estimate*). Terdapat beberapa metode yang telah dikembangkan untuk menaksir parameter model, yang paling populer adalah: *Least square* (LS), *Maximum likelihood* (ML), dan *Bayesian methods* (Shaqlaih, 2010).

Teori *Maximum likelihood* merupakan salah satu metode yang digunakan untuk mengestimasi model. Model yang baik menurut teori ini diperoleh bila *likelihood* mencapai nilai maksimum atau dikenal dengan maksimum *likelihood*. Maksimum *likelihood* merupakan salah satu pengembangan teori statistik yang sangat penting di abad 20. Metode ini merupakan suatu karya besar R.A. Fisher yang dikembangkan di antara tahun 1912 dan 1922 (Aldrich, 1997).

Shaqlaih (2010) dalam disertasinya menyajikan definisi yang terkait dengan fungsi *likelihood* yaitu:

Definisi 2.19. (Shaqlaih, 2010)

Misalkan X_1, \dots, X_n adalah n variabel acak yang memiliki fungsi pdf $f_i(x_i, \theta)$ yang bergantung pada parameter θ , maka p.d.f bersama dari n pengamatan yang saling bebas $x = (x_1, \dots, x_n)^T$ adalah $f(x|\theta) = \prod_{i=1}^n f_i(x_i, \theta)$. Maka fungsi $\prod_{i=1}^n f_i(x_i, \theta)$ adalah fungsi *likelihood* dalam θ , atau ditulis $L(\theta|x)$ yang merupakan suatu fungsi dengan parameter θ bila data x diketahui.

Sedangkan Metode Bayesian, menurut Walpole (1992) adalah:

“Metode yang menggunakan atau menggabungkan pengetahuan subjektif (terdahulu) tentang parameter model yang akan ditaksir dengan informasi yang diperoleh dari data sampel. Informasi terdahulu biasa disebut dengan informasi *prior*, yang diperoleh dari distribusi parameter tersebut. Informasi dari data dirangkum dalam fungsi *likelihood*. Penggabungan dari informasi *prior* dan informasi dari data akan menghasilkan informasi *posterior*. Selanjutnya penarikan kesimpulan mengenai parameter model yang ditaksir berdasarkan pada distribusi *posterior*.” Merujuk pada pendapat Walpole (1992) tersebut, maka tahap pertama yang harus dilakukan dalam penelitian adalah mendefinisikan model.

Menurut Myung (2003) jika model itu sudah didefinisikan dan data sudah tersedia maka langkah yang dilakukan selanjutnya adalah mengevaluasi *goodness of fit*, yaitu ukuran statistik yang menunjukkan seberapa cocok model yang

terbentuk dengan data observasi. *Goodness of fit* dapat diperoleh dengan menganalisa kecocokan model.

Demikian juga menurut Alpaydin (2004) pemilihan model dengan metode *Bayesian* dapat diterapkan bila modelnya memiliki informasi *prior*. Informasi *prior* ini didefinisikan dengan fungsi distribusi *prior* dari model, yaitu $p(\text{model})$. Bila data diberikan dan modelnya ada, maka *posterior* $p(\text{model}|\text{data})$ dapat dihitung dengan menggunakan teorema Bayes.

Informasi *posterior* dapat diperoleh dengan memanfaatkan informasi *prior*, yang kemudian informasi dari *posterior* tersebut akan dimanfaatkan kembali untuk memperbaiki informasi *prior*. Semua hubungan baik *posterior*, *prior* dan *likelihood* dari model yang akan dibangun dapat disajikan dalam tabel berikut ini:

Tabel 2-2 Model Bayesian : *Posterior, Likelihood dan Prior*
(Alpaydin, 2004)

$P(\text{model} \text{data}) = \frac{P(\text{data} \text{model})P(\text{model})}{P(\text{data})}$		
$P(\text{model} \text{data})$	$P(\text{data} \text{model})$	$P(\text{model})$
<i>posterior</i>	<i>likelihood</i>	<i>prior</i>

BAB 3

FAKTORISASI MATRIKS PROBABILITAS

Bab ini berisi tentang proses yang terdapat dalam teori faktorisasi matriks. Semuanya terbagi kedalam empat subbab, subbab pertama membahas faktorisasi matriks secara umum, subbab kedua membahas faktorisasi matriks probabilitas, subbab ketiga membahas algoritma yang terkait dengan metode faktorisasi matriks probabilitas dan subbab keempat berisi tentang ilustrasi dan simulasi CF pada data dalam skala kecil.

3.1. Faktorisasi Matriks

Saat ini teknik faktorisasi matriks merupakan metode yang banyak dipakai dalam sistem rekomendasi. Pada prinsipnya metode ini hampir sama dengan metode-metode *Principal Component Analysis (PCA)* dan *Singular Value Decomposition (SVD)* (Graus, 2011).

Dalam bab satu sudah dijelaskan bahwa masalah dalam faktorisasi matriks ini adalah jika diberikan suatu matriks $R_{n \times m}$ bagaimana mendapatkan matriks $W_{n \times k}$ dan matriks $H_{k \times m}$ dimana $k \ll (n, m)$ agar hasil perkalian dari kedua matriks tersebut memiliki selisih yang sekecil mungkin bila dibandingkan dengan matriks $R_{n \times m}$ atau ditulis $WH \approx R$. Misalkan matriks-matriks tersebut adalah :

$$W = \begin{bmatrix} w_{11} & \dots & w_{1k} \\ \dots & \dots & \dots \\ w_{n1} & \dots & w_{nk} \end{bmatrix} \quad H = \begin{bmatrix} h_{11} & \dots & h_{1m} \\ \dots & \dots & \dots \\ h_{k1} & \dots & h_{km} \end{bmatrix} \quad R = \begin{bmatrix} r_{11} & \dots & r_{1m} \\ \dots & \dots & \dots \\ r_{n1} & \dots & r_{nm} \end{bmatrix}$$

Dalam matriks ini misalkan ada vektor w_i dan h_j dimana : w_i adalah vektor yang entri-entrinya merupakan elemen dari baris ke i matriks W , demikian juga h_j adalah vektor yang entri-entrinya merupakan elemen dari kolom ke j matriks H atau ditulis :

$$w_i = \begin{bmatrix} w_{i1} \\ w_{i,2} \\ \vdots \\ w_{i,k} \end{bmatrix} \quad \text{dan} \quad h_j = \begin{bmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{kj} \end{bmatrix} \quad (3.1)$$

Misalkan perkalian matriks W dan H adalah $WH = \hat{R}$ dimana :

$$\hat{R} = \begin{bmatrix} \hat{r}_{11} & \dots & \hat{r}_{1m} \\ \dots & \dots & \dots \\ \hat{r}_{n1} & \dots & \hat{r}_{nm} \end{bmatrix}$$

maka entri matriks \hat{R} pada posisi baris ke- i kolom ke- j merupakan hasil perkalian dua vektor \mathbf{w}_i dan \mathbf{h}_j atau ditulis $\hat{r}_{ij} = \mathbf{w}_i^T \mathbf{h}_j$. Semakin kecil perbedaan antara skor rating prediksi \hat{r}_{ij} dengan skor rating aktual r_{ij} pada data testing, maka pemilihan matriks W dan H dalam model faktorisasi matriks akan semakin baik. Dengan demikian model yang baik adalah model yang memenuhi optimasi berikut ini:

$$\min_{\mathbf{w}_i, \mathbf{h}_j} f, \text{ dimana } f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} (r_{ij} - \mathbf{w}_i^T \mathbf{h}_j)^2 \quad (3.2)$$

3.2. Faktorisasi Matriks Probabilitas

Perbedaan faktorisasi matriks probabilitas dengan faktorisasi matriks pada subbab 3.1 terletak pada kriteria pemilihan data untuk entri-entri dari matriks W dan H . Dalam faktorisasi matriks, pemilihan matriks W dan H bisa sembarang, namun untuk faktorisasi matriks probabilitas tidak bisa berlaku sembarang. Faktorisasi matriks probabilitas, mensyaratkan entri-entri dalam matriks W dan H merupakan variabel acak yang diasumsikan berdistribusi normal multivariat berdimensi k .

Salakhutdinov & Mnih (2008) mendefinisikan probabilitas bersyarat untuk seluruh skor rating yang diamati dengan:

$$P(R|W, H, \sigma^2) = \prod_{i=1}^N \prod_{j=1}^M [\mathcal{N}(r_{i,j} | \mathbf{w}_i^T \mathbf{h}_j, \sigma^2)]^{I_{i,j}} \quad (3.3)$$

dimana $I_{i,j}$ adalah fungsi indikator yang didefinisikan:

$$I_{i,j} = \begin{cases} 1, & \text{jika pengguna } i \text{ merating film } j \\ 0, & \text{jika pengguna } i \text{ tidak memberikan rating film } j \end{cases}$$

Menurut Koch (2007) pada definisi 2.18 untuk vektor w_i yang berdimensi k dan berdistribusi normal $\mathcal{N}_k(\mu, \Sigma_w)$ dimana μ adalah vektor mean yang berukuran $k \times 1$ dan matriks varian kovarian Σ berukuran $k \times k$ yang definit positif, maka $P(w_i | \mu, \Sigma_w)$ dapat dinyatakan dengan:

$$P(w_i | \mu, \Sigma_w) = \frac{1}{(2\pi)^{\frac{k}{2}} |\Sigma_w|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(w_i - \mu)^T \Sigma^{-1} (w_i - \mu)\right)} \quad (3.4)$$

dimana vektor $w_i = (w_{i1}, w_{i2}, \dots, w_{ik})^T$ dapat dianalogikan dengan vektor acak yang diperoleh dari pengamatan ke i dari data yang memiliki k variabel. Oleh karena itu jika dilakukan n buah pengamatan dari populasi tersebut maka deretan pengamatan dalam penelitian ini dapat dinyatakan dengan vektor :

$w_1 = (w_{11}, w_{12}, \dots, w_{1k})^T$, $w_2 = (w_{21}, w_{22}, \dots, w_{2k})^T$, sampai dengan $w_n = (w_{n1}, w_{n2}, \dots, w_{nk})^T$ maka penulisan secara umum untuk pengamatan ke- i dapat dinyatakan dengan vektor :

$$w_i = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{ik} \end{bmatrix} \quad (3.5)$$

Sedangkan mean untuk semua pengamatan tersebut adalah vektor:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{bmatrix} \quad (3.6)$$

Bilamana semua pengamatan itu digabungkan, maka himpunan dari semua pengamatan tersebut dapat dituliskan dalam bentuk matriks W berikut :

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1k} \\ w_{21} & w_{22} & \dots & w_{2k} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nk} \end{bmatrix} \quad (3.7)$$

dimana jumlah kolomnya menyatakan banyaknya variabel dari setiap data yang diamati, sedangkan banyaknya baris menyatakan jumlah keseluruhan data yang diamati. Menurut Rencher (2002), matriks (3.7) adalah bentuk matriks yang menyajikan n sampel data multivariat berdimensi k . Sehingga nilai varian dan kovariannya dapat ditulis dalam matriks berikut :

$$\Sigma_w = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1k} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2k} \\ \dots & \dots & \dots & \dots \\ \sigma_{k1} & \sigma_{k2} & \dots & \sigma_{kk} \end{bmatrix} \quad (3.8)$$

Entri-entri diagonal pada matriks Σ_w merupakan varian, sedangkan entri selain diagonal utamanya adalah kovarian. Bilamana dalam pengamatan tersebut semua variabelnya saling bebas (*independent*), maka entri σ_{ij} dalam matriks (3.8) sama dengan nol atau ditulis $\sigma_{ij} = 0$ untuk $\forall i \neq j$, sehingga matriks kovarian Σ_w pada persamaan (3.8) dapat ditulis kembali menjadi matriks diagonal sbb:

$$\Sigma_w = \begin{bmatrix} \sigma_{11} & 0 & \dots & 0 \\ 0 & \sigma_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_{kk} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_k^2 \end{bmatrix} \quad (3.9)$$

Dengan mengasumsikan $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2 = \sigma^2$, maka persamaan (3.9) dapat disederhanakan menjadi:

$$\Sigma_w = \sigma_w^2 \mathbf{I} \quad (3.10)$$

dimana \mathbf{I} adalah matriks identitas yang berukuran $k \times k$, sedangkan σ_w^2 adalah nilai varian σ^2 berasal dari matriks Σ_w , maka determinan dari matriks Σ_w dapat ditulis dengan:

$$|\Sigma_w| = (\sigma_w^2)^k \text{ sedangkan } \Sigma_w^{-1} = \begin{bmatrix} 1/\sigma_w^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_w^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1/\sigma_w^2 \end{bmatrix}$$

Pada persamaan (3.4) bisa dilihat bahwa bahwa jarak Mahalanobis dari w_i ke μ adalah :

$$\Delta(w_i) = \frac{1}{2} (w_i - \mu)^T \Sigma_w^{-1} (w_i - \mu)$$

Untuk model yang berbentuk *zero mean Gaussian*, dipilih $\mu = \mathbf{0}$, sehingga

jarak Mahalanobis dari w_i ke μ dapat dijabarkan kembali menjadi:

$$\Delta(w_i) = \frac{1}{2} (w_i - \mu)^T \Sigma_w^{-1} (w_i - \mu)$$

$$\begin{aligned}
&= \frac{1}{2} (\mathbf{w}_i)^T \Sigma_{\mathbf{W}}^{-1} (\mathbf{w}_i) \\
&= \frac{1}{2} [w_{i1} \quad w_{i,2} \quad \dots \quad w_{i,k}] \begin{bmatrix} 1/\sigma_{\mathbf{W}}^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_{\mathbf{W}}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_{\mathbf{W}}^2 \end{bmatrix} \begin{bmatrix} w_{i1} \\ w_{i,2} \\ \vdots \\ w_{i,k} \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} w_{i1} & w_{i2} & \dots & w_{ik} \\ \frac{1}{\sigma_{\mathbf{W}}^2} & \frac{1}{\sigma_{\mathbf{W}}^2} & \dots & \frac{1}{\sigma_{\mathbf{W}}^2} \end{bmatrix} \begin{bmatrix} w_{i1} \\ w_{i,2} \\ \vdots \\ w_{i,k} \end{bmatrix} \\
&= \frac{1}{2\sigma_{\mathbf{W}}^2} \left(\sum_{s=1}^k w_{i,s}^2 \right) \tag{3.11}
\end{aligned}$$

Cara yang sama berlaku juga untuk matriks H, sehingga matriks varian kovarian $\Sigma_{\mathbf{H}}$, dapat dinyatakan dengan :

$$\Sigma_{\mathbf{H}} = \sigma_{(\mathbf{H})}^2 \mathbf{I} \tag{3.12}$$

maka:

$$|\Sigma_{\mathbf{H}}| = (\sigma_{\mathbf{H}}^2)^k \text{ dan } \Sigma_{\mathbf{H}}^{-1} = \begin{bmatrix} 1/\sigma_{\mathbf{H}}^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_{\mathbf{H}}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_{\mathbf{H}}^2 \end{bmatrix}$$

sehingga

$$\begin{aligned}
\Delta(\mathbf{h}_j) &= \frac{1}{2} (\mathbf{h}_j - \boldsymbol{\mu})^T \Sigma_{\mathbf{H}}^{-1} (\mathbf{h}_j - \boldsymbol{\mu}) \\
&= \frac{1}{2} (\mathbf{h}_j)^T \Sigma_{\mathbf{H}}^{-1} (\mathbf{h}_j), \text{ karena } \boldsymbol{\mu} = \mathbf{0} \\
&= \frac{1}{2} [h_{1j} \quad h_{2j} \quad \dots \quad h_{kj}] \begin{bmatrix} 1/\sigma_{\mathbf{H}}^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_{\mathbf{H}}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_{\mathbf{H}}^2 \end{bmatrix} \begin{bmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{kj} \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} h_{1j} & h_{2j} & \dots & h_{kj} \\ \frac{1}{\sigma_{\mathbf{H}}^2} & \frac{1}{\sigma_{\mathbf{H}}^2} & \dots & \frac{1}{\sigma_{\mathbf{H}}^2} \end{bmatrix} \begin{bmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{kj} \end{bmatrix}
\end{aligned}$$

$$= \frac{1}{2\sigma_H^2} \left(\sum_{s=1}^k h_{sj}^2 \right) \quad (3.13)$$

sedangkan $P(\mathbf{h}_j | \boldsymbol{\mu}, \boldsymbol{\Sigma}_H)$ menurut Koch (2007) adalah:

$$P(\mathbf{h}_j | \boldsymbol{\mu}, \boldsymbol{\Sigma}_H) = \frac{1}{(2\pi)^{\frac{k}{2}} |\boldsymbol{\Sigma}_H|^{\frac{1}{2}}} e^{\left(-\frac{1}{2} (\mathbf{h}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_H^{-1} (\mathbf{h}_j - \boldsymbol{\mu}) \right)} \quad (3.14)$$

Demikian juga menurut Dueck & Frey (2004) dalam Salakhutdinov & Mnih (2008) disebutkan bahwa untuk probabilitas *prior Gaussian spherical* dari vektor fitur pengguna dan vektor fitur film dapat dinyatakan dengan persamaan (3.15) dan (3.16) berikut :

$$\begin{aligned} P(W | \sigma_W^2) &= \prod_{i=1}^N \mathcal{N}(\mathbf{w}_i | 0, \sigma_W^2 \mathbf{I}) \\ &= \prod_{i=1}^N P(\mathbf{w}_i | 0, \sigma_W^2 \mathbf{I}) \\ &= \prod_{i=1}^N \frac{1}{(2\pi)^{\frac{k}{2}} |\boldsymbol{\Sigma}_W|^{\frac{1}{2}}} e^{\left(-\frac{1}{2} (\mathbf{w}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_W^{-1} (\mathbf{w}_i - \boldsymbol{\mu}) \right)} \\ &= \prod_{i=1}^N \frac{1}{(2\pi)^{\frac{k}{2}} (\sigma_W^2)^{\frac{k}{2}}} e^{-\frac{1}{2\sigma_W^2} (\sum_{s=1}^k w_{i,s}^2)} \\ &= \frac{1}{(2\pi)^{kN/2} (\sigma_W^2)^{kN/2}} e^{-\frac{1}{2\sigma_W^2} (\sum_{i=1}^N \sum_{s=1}^k w_{i,s}^2)} \\ &= \frac{1}{(2\pi)^{kN/2} (\sigma_W^2)^{kN/2}} e^{-\frac{1}{2\sigma_W^2} \|\mathbf{W}\|_F^2} \end{aligned} \quad (3.15)$$

$$P(H | \sigma_H^2) = \prod_{j=1}^M \mathcal{N}(\mathbf{h}_j | 0, \sigma_H^2 \mathbf{I})$$

$$\begin{aligned}
&= \prod_{j=1}^M P(h_j | 0, \sigma_{(H)}^2 \mathbf{I}) \\
&= \prod_{j=1}^M \frac{1}{(2\pi)^{\frac{k}{2}} |\Sigma_H|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(\mathbf{h}_j - \mu)^T \Sigma_H^{-1} (\mathbf{h}_j - \mu)\right)} \\
&= \prod_{j=1}^M \frac{1}{(2\pi)^{\frac{k}{2}} (\sigma_H^2)^{\frac{k}{2}}} e^{-\frac{1}{2\sigma_H^2} (\sum_{s=1}^k h_{s,j}^2)} \\
&= \frac{1}{(2\pi)^{kM/2} (\sigma_H^2)^{kM/2}} e^{-\frac{1}{2\sigma_H^2} (\sum_{j=1}^M \sum_{s=1}^k h_{s,j}^2)} \\
&= \frac{1}{(2\pi)^{kM/2} (\sigma_H^2)^{kM/2}} e^{-\frac{1}{2\sigma_H^2} \|\mathbf{H}\|_F^2} \tag{3.16}
\end{aligned}$$

Dalam persamaan (3.15) dan (3.16) tersebut,

$\|\mathbf{W}\|_F^2 = (\sum_{i=1}^N \sum_{s=1}^k w_{i,s}^2)$ adalah norm Frobenius dari matriks \mathbf{W}

$\|\mathbf{H}\|_F^2 = (\sum_{j=1}^M \sum_{s=1}^k h_{s,j}^2)$ adalah norm Frobenius dari matriks \mathbf{H}

Berdasarkan persamaan (3.15) dan (3.16), matriks \mathbf{W} dan \mathbf{H} dapat diperoleh dengan menggunakan estimasi Bayesian seperti yang sudah dijelaskan dalam subbab 2.5 Tabel 2-2 yaitu dengan persamaan:

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}$$

Menurut Lim & Teh (2007) dalam faktorisasi matriks, persoalannya adalah bagaimana mengaproksimasi *posterior* persamaan (3.17) berikut:

$$P(\mathbf{W}, \mathbf{H} | \mathbf{R}, \sigma^2, \sigma_W^2, \sigma_H^2) = \frac{P(\mathbf{R} | \mathbf{W}, \mathbf{H}, \sigma^2) P(\mathbf{W} | \sigma_W^2) P(\mathbf{H} | \sigma_H^2)}{P(\mathbf{R} | \sigma^2)} \tag{3.17}$$

Cara yang paling sederhana untuk mengaproksimasi *posterior* adalah dengan terlebih dahulu menguraikan masing-masing komponen sebelah kanan dari

persamaan (3.17) menjadi : $P(R|W, H, \sigma^2)$, $P(W|\sigma_W^2)$, $P(H|\sigma_H^2)$ dan $P(R|\sigma^2)$ dimana perkalian $P(W|\sigma_W^2)$ dan $P(H|\sigma_H^2)$ menghasilkan *prior*:

$$\begin{aligned} P(W|\sigma_W^2)P(H|\sigma_H^2) &= \prod_{i=1}^N \prod_{j=1}^M (\mathcal{N}(w_i|0, \sigma_W^2 \mathbf{I})) \mathcal{N}(h_j|0, \sigma_H^2 \mathbf{I}) \\ &= \frac{1}{(2\pi\sigma_W^2)^{kN/2}} e^{-\frac{1}{2\sigma_W^2} \|W\|_F^2} \frac{1}{(2\pi\sigma_H^2)^{kM/2}} e^{-\frac{1}{2\sigma_H^2} \|H\|_F^2} \end{aligned} \quad (3.18)$$

sedangkan *likelihood* nya adalah:

$$P(R|W, H, \sigma^2) = \prod_{i=1}^N \prod_{j=1}^M \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_{i,j} - w_i^T \cdot h_j)^2}{2\sigma^2}} \right)^{I_{i,j}} = \mathcal{L}(W, H, \sigma^2|R) \quad (3.19)$$

dan

$$P(R|\sigma^2) = \prod_{i=1}^N \prod_{j=1}^M \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_{i,j})^2}{2\sigma^2}} \right)^{I_{i,j}} \quad (3.20)$$

Hasil penguraian pada persamaan (3.18), (3.19), dan (3.20) kemudian disubstitusikan ke persamaan (3.17). Maka log posterior persamaan (3.17) menjadi :

$$\begin{aligned} \ln(P(W, H|R, \sigma^2, \sigma_W^2, \sigma_H^2)) &= \\ &= \ln \left(\frac{P(R|W, H, \sigma^2)P(W|\sigma_W^2)P(H|\sigma_H^2)}{P(R|\sigma^2)} \right) \\ &= \ln(P(R|W, H, \sigma^2)) + \ln(P(W|\sigma_W^2)) + \ln(P(H|\sigma_H^2)) - \ln(P(R|\sigma^2)) \end{aligned} \quad (3.21)$$

sedangkan:

$$\ln(P(R|W, H, \sigma^2)) = \ln \left(\prod_{i=1}^N \prod_{j=1}^M \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_{i,j} - w_i^T \cdot h_j)^2}{2\sigma^2}} \right)^{I_{i,j}} \right)$$

$$= C1 - \left(\sum_{i=1}^N \sum_{j=1}^M I_{ij} \right) \ln \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2$$

$$\ln(P(W|\sigma_W^2)) = \ln \left(\frac{1}{(2\pi)^{kN/2} (\sigma_W^2)^{kN/2}} e^{-\frac{1}{2\sigma_W^2} \|W\|_F^2} \right)$$

$$= C2 - \frac{1}{2} kN \ln \sigma_W^2 - \frac{1}{2\sigma_W^2} \|W\|_F^2$$

$$\ln(P(H|\sigma_H^2)) = \frac{1}{(2\pi)^{kM/2} (\sigma_H^2)^{kM/2}} e^{-\frac{1}{2\sigma_H^2} \|H\|_F^2}$$

$$= C3 - \frac{1}{2} kM \ln \sigma_H^2 - \frac{1}{2\sigma_H^2} \|H\|_F^2$$

$$\ln(P(R|\sigma^2)) = \prod_{i=1}^N \prod_{j=1}^M \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_{ij})^2}{2\sigma^2}} \right)^{I_{i,j}}$$

$$= C4$$

maka secara keseluruhan

$$\ln(p(W, H|R, \sigma^2, \sigma_W^2, \sigma_H^2))$$

$$= -\frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 - \frac{1}{2\sigma_W^2} \|W\|_F^2 - \frac{1}{2\sigma_H^2} \|H\|_F^2$$

$$- \frac{1}{2} \left(\sum_{i=1}^N \sum_{j=1}^M I_{ij} \right) \ln \sigma^2 - \frac{1}{2} kN \ln \sigma_W^2 - \frac{1}{2} kM \ln \sigma_H^2 + C1 + C2 + C3 + C4$$

$$= -\left(\frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \frac{1}{2\sigma_W^2} \sum_{i=1}^N \mathbf{w}_i^T \mathbf{w}_i + \frac{1}{2\sigma_H^2} \sum_{j=1}^M \mathbf{h}_j^T \mathbf{h}_j \right)$$

$$- \frac{1}{2} \left(\left(\sum_{i=1}^N \sum_{j=1}^M I_{ij} \right) \ln \sigma^2 + \frac{1}{2} kN \ln \sigma_W^2 + \frac{1}{2} kM \ln \sigma_H^2 \right) + C$$

$$= -B(\mathbf{w}_i, \mathbf{h}_j) - \frac{1}{2} \left(\left(\sum_{i=1}^N \sum_{j=1}^M I_{ij} \right) \ln \sigma^2 + \frac{1}{2} kN \ln \sigma_W^2 + \frac{1}{2} kM \ln \sigma_H^2 \right) + C \quad (3.22)$$

dimana :

$$B(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (r_{ij} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \frac{1}{2\sigma_W^2} \sum_{i=1}^N \mathbf{w}_i^T \mathbf{w}_i + \frac{1}{2\sigma_H^2} \sum_{j=1}^M \mathbf{h}_j^T \mathbf{h}_j$$

adalah komponen persamaan yang dinyatakan dalam $\mathbf{w}_i, \mathbf{h}_j$ sedangkan $C=C1+C2+C3+C4$ adalah konstanta bilangan real.

Lim & Teh (2007) dalam tulisannya menyebutkan bahwa dalam penggunaan metode estimasi Bayesian, pemilihan W dan H didasarkan pada nilai posterior maksimum dari persamaan (3.17) :

$$\max_{W,H} [f(W,H) = \frac{P(R|W,H,\sigma^2)P(W|\sigma_W^2)P(H|\sigma_H^2)}{P(R|\sigma^2)}] \quad (3.23)$$

Metode yang sama juga disampaikan Salakhutdinov & Mnih (2008) bahwa untuk memaksimumkan log posterior $P(W,H|R,\sigma^2,\sigma_W^2,\sigma_H^2)$ dari persamaan (3.22) hasilnya akan ekuivalen dengan meminimumkan $B(\mathbf{w}_i, \mathbf{h}_j)$. Oleh karena itu untuk menentukan log posterior maksimum $P(W,H|R,\sigma^2,\sigma_W^2,\sigma_H^2)$ sebaiknya terlebih dahulu dilakukan analisa pada komponen $B(\mathbf{w}_i, \mathbf{h}_j)$.

Andaikan $B(\mathbf{w}_i, \mathbf{h}_j)$ dikalikan dengan σ^2 maka diperoleh :

$$\sigma^2 B(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (r_{ij} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \frac{\sigma^2}{2\sigma_W^2} \|\mathbf{W}\|_F^2 + \frac{\sigma^2}{2\sigma_H^2} \|\mathbf{H}\|_F^2$$

misalkan $\lambda_W = \frac{\sigma^2}{\sigma_W^2}$, $\lambda_H = \frac{\sigma^2}{\sigma_H^2}$, dan $\sigma^2 B(\mathbf{w}_i, \mathbf{h}_j) = f(\mathbf{w}_i, \mathbf{h}_j)$ maka:

$$f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (r_{ij} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \frac{\lambda_W}{2} \sum_{i=1}^N \mathbf{w}_i^T \mathbf{w}_i + \frac{\lambda_H}{2} \sum_{j=1}^M \mathbf{h}_j^T \mathbf{h}_j \quad (3.24)$$

Bila λ_W dan λ_H pada persamaan (3.24) diasumsikan nilainya sama, yaitu $\lambda_W = \lambda_H = \lambda$, maka persamaan (3.24) dapat disederhanakan menjadi:

$$f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \frac{\lambda}{2} \left(\sum_{i=1}^N \mathbf{w}_i^T \mathbf{w}_i + \sum_{j=1}^M \mathbf{h}_j^T \mathbf{h}_j \right) \quad (3.25)$$

Persamaan (3.25) tiada lain adalah *loss function*, yaitu fungsi yang menyatakan perbedaan skor *rating* aktual dengan *rating* prediksi pada pengamatan ke (i,j) dari sampel data yang diamati. Model yang baik adalah model yang memiliki *loss* yang sekecil mungkin. Dalam simulasi, salah satu ukuran yang digunakan untuk menyatakan akurasi model adalah RMSE.

Oleh karena itu model yang baik adalah model yang memiliki nilai RMSE yang kecil. Menurut Su & Khoshgoftaar (2009) yang diuraikan pada subbab 4.5 komponen $(r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2$ pada persamaan (3.25) ekuivalen dengan RMSE. Dengan demikian model terbaik terjadi bila \mathbf{w}_i dan \mathbf{h}_j memenuhi fungsi optimasi berikut:

$$\min_{\mathbf{w}_i, \mathbf{h}_j} f(\mathbf{w}_i, \mathbf{h}_j) \quad (3.26)$$

dimana :

$$f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \frac{\lambda}{2} \left(\sum_{i=1}^N \mathbf{w}_i^T \mathbf{w}_i + \sum_{j=1}^M \mathbf{h}_j^T \mathbf{h}_j \right)$$

komponen $\sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2$ pada persamaan (3.26) tersebut dapat disederhanakan kedalam bentuk norm Frobenius $\|\cdot\|_F^2$.

Untuk mengubah bentuk : $\sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2$ menjadi norm Frobenius, menurut Lakshminarayanan, Bouchard, & Cedric (2011) ada matriks $X \in \mathbb{R}^{N \times M}$ yang merupakan matriks indikator dari matriks R , yaitu matriks X yang entri-entrinya adalah:

$$x_{i,j} = \begin{cases} 1, & \text{Jika entri matrik } r_{i,j} \neq 0 \\ 0, & \text{Sebaliknya} \end{cases}$$

Maka bentuk $\sum_{i=1}^N \sum_{j=1}^M I_{i,j} (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2$ dapat dituliskan lebih sederhana lagi menjadi $\|X \odot (R - WH)\|_F^2$, dimana \odot adalah perkalian Hadamard, maka persamaan (3.25) dapat dirubah menjadi :

$$f(W, H) = \frac{1}{2} \left(\|X \odot (R - WH)\|_F^2 + \lambda (\|W\|_F^2 + \|H\|_F^2) \right) \quad (3.27)$$

Demikian juga dengan fungsi optimasi (3.26.), bentuknya dapat dituliskan menjadi:

$$\min_{W, H} f(W, H) \quad (3.28)$$

dimana :

$$f(W, H) = \frac{1}{2} \left(\|X \odot (R - WH)\|_F^2 + \lambda (\|W\|_F^2 + \|H\|_F^2) \right)$$

$f(W, H)$ minimum (3.28) ekuivalen dengan nilai RMSE minimum. Menurut Pilaszy (2009), metoda *incremental gradient descent* dapat digunakan untuk mendapatkan nilai lokal minimum dari fungsi $f(W, H)$. Dengan metoda tersebut maka secara komputasi penyelesaian masalah optimasi berikut :

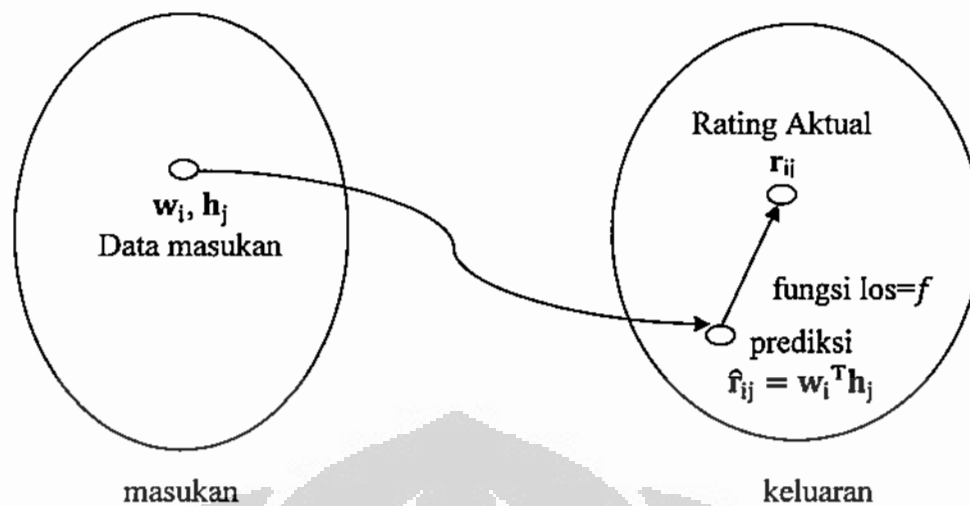
$$\min_{W, H} f(W, H)$$

dapat direalisasikan ke dalam algoritma *incremental gradient descent*.

Ilustrasi proses penentuan nilai minimum dari fungsi $f(W, H)$ bisa dilakukan dengan terlebih dahulu mengambil satu skor *rating* ke (i, j) dari sample data *rating* yang diamati. Setiap data pengamatan dapat mengandung *loss* baik besar maupun kecil. Besarnya *loss* pada rating ke (i, j) dapat diketahui dari fungsi *loss* berikut:

$$f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \left((r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda \mathbf{w}_i^T \mathbf{w}_i + \lambda \mathbf{h}_j^T \mathbf{h}_j \right) \quad (3.29)$$

Dalam kaidah *machine learning*, keterkaitan *input*, *output*, *loss* dan prediksi ini dapat digambarkan dengan skema seperti yang terlihat pada Gambar 3.1 berikut:



Gambar 3-1 *Framework Machine Learning* untuk aplikasi Faktorisasi Matriks Probabilitas (Liu, 2011)

Masalah optimasi dalam penentuan nilai minimum *loss function* dapat ditentukan dengan mencari nilai minimum fungsi $f(\mathbf{w}_i, \mathbf{h}_j)$, yaitu dengan menurunkan fungsi tersebut terhadap \mathbf{w}_i dan \mathbf{h}_j .

diketahui:
$$f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \left((r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda \mathbf{w}_i^T \mathbf{w}_i + \lambda \mathbf{h}_j^T \mathbf{h}_j \right)$$

maka
$$\frac{\partial f}{\partial \mathbf{w}_i} = \frac{1}{2} \frac{\partial}{\partial \mathbf{w}_i} \left((r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda \mathbf{w}_i^T \mathbf{w}_i + \lambda \mathbf{h}_j^T \mathbf{h}_j \right)$$

$$= -(r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j) \mathbf{h}_j + \lambda \mathbf{w}_i = -e_{ij} \mathbf{h}_j + \lambda \mathbf{w}_i$$

sehingga
$$\frac{\partial f}{\partial \mathbf{w}_i} = -e_{ij} \mathbf{h}_j + \lambda \mathbf{w}_i \quad (3.30)$$

dengan cara yang sama dengan proses penurunan pada persamaan (3.30), maka turunan f terhadap \mathbf{h}_j adalah:

$$\frac{\partial f}{\partial \mathbf{h}_j} = \frac{1}{2} \frac{\partial}{\partial \mathbf{h}_j} \left((r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda \mathbf{w}_i^T \mathbf{w}_i + \lambda \mathbf{h}_j^T \mathbf{h}_j \right)$$

$$= -(r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j) \mathbf{w}_i + \lambda \mathbf{h}_j = -e_{ij} \mathbf{w}_i + \lambda \mathbf{h}_j$$

sehingga
$$\frac{\partial f}{\partial \mathbf{h}_j} = -e_{ij} \mathbf{w}_i + \lambda \mathbf{h}_j \quad (3.31)$$

Pada persamaan (3.30) dan (3.31), e_{ij} adalah bentuk penyederhanaan dari penulisan $(r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)$ yaitu besarnya *error* pada titik pengamatan ke (i,j) .

Error tersebut merupakan selisih skor *rating* aktual dengan nilai prediksi untuk satu titik pengamatan ke (i, j) .

Pilaszy (2009) menyebutkan bahwa bobot w_i dan h_j , akan terus diperbaharui melalui proses iterasi dan akan mencapai nilai optimal saat RMSE minimum pada data *testing*. Proses memperbaharui w_i dan h_j dapat ditunjukkan dengan algoritma berikut ini:

update vektor w_i

$$w_i \leftarrow w_i + \eta \left(-\frac{\partial f}{\partial w_i} \right) \text{ dimana } \frac{\partial f}{\partial w_i} \text{ diambil dari persamaan (3.30)}$$

$$\text{maka : } w_i \leftarrow w_i + \eta(e_{ij} h_j - \lambda w_i)$$

update vektor h_j

$$h_j \leftarrow h_j + \eta \left(-\frac{\partial f}{\partial h_j} \right) \text{ dimana } \frac{\partial f}{\partial h_j} \text{ diambil dari persamaan (3.31)}$$

$$\text{maka : } h_j \leftarrow h_j + \eta(e_{ij} w_i - \lambda h_j)$$

Dalam hal ini η adalah *learning rate* yaitu parameter simulasi yang merupakan koefisien laju pembelajaran $\frac{\partial f}{\partial w_i}$ dan $\frac{\partial f}{\partial h_j}$.

3.3. Algoritma Faktorisasi Matriks Probabilitas

Berikut adalah beberapa variabel dan data yang digunakan dalam algoritma faktorisasi matriks probabilitas.

Data *rating* (D) ;

Data D dibagi menjadi dua bagian yang *disjoint*

1 : \mathcal{R} : Data Training

2 : \mathcal{V} : Data Testing

λ : Parameter regularisasi η : *learning rate* dan k : jumlah fitur

Inisialisasi matriks W dan H

Algoritma 3-1: Algoritma Faktorisasi Matriks Probabilitas

Iterasi 1 sd max iterasi

Baca (\mathcal{R} : Data Training):

Load (u.base):

proses untuk \forall elemen $(q, (i, j, r_{ij})) \in \mathcal{R}$ yang terurut berdasarkan q

$$e_{i,j} = r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j$$

$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \eta(e_{i,j} \mathbf{h}_j - \lambda \mathbf{w}_i)$$

$$\mathbf{h}_j \leftarrow \mathbf{h}_j + \eta(e_{i,j} \mathbf{w}_i - \lambda \mathbf{h}_j)$$

Akhir proses elemen $(q, (i, j, r_{ij})) \in \mathcal{R}$

$$\mathbf{W} = \text{vertcat}(\mathbf{w}_i^T)$$

$$\mathbf{H} = \text{horzcat}(\mathbf{h}_j)$$

untuk \forall elemen $(q, (i, j, r_{ij})) \in \mathcal{R}$

Gunakan persamaan (3.29) untuk menghitung error pada data training

$$\text{SE}(\text{Training}[i, j]) = (1/2)((r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda \mathbf{w}_i^T \mathbf{w}_i + \lambda \mathbf{h}_j^T \mathbf{h}_j)$$

save($q, i, j, \mathbf{w}_i, \mathbf{h}_j, \text{SE}(\text{Training}[i, j])$)

Akhir proses elemen $(q, (i, j, r_{ij})) \in \mathcal{R}$

$$\text{RMSE}(\text{iterasi}, \text{Training}) = \left(\frac{1}{|\mathcal{R}|} \sum_{(i,j,r_{i,j}) \in \mathcal{R}} \text{SE}(\text{Training}[i, j]) \right)^{1/2}$$

Baca (\mathcal{V} : Data Testing):

Load(u.test):

proses untuk \forall elemen $(q, (i, j, r_{ij})) \in \mathcal{V}$ yang terurut berdasarkan q

$$\text{SE}(\text{Testing}[i, j]) = (r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2$$

save($q, i, j, \mathbf{w}_i, \mathbf{h}_j, \text{SE}(\text{Testing}[i, j])$)

Akhir proses elemen $(q, (i, j, r_{ij})) \in \mathcal{V}$

$$\text{RMSE}(\text{iterasi}, \text{Testing}) = \left(\frac{1}{|\mathcal{V}|} \sum_{(i,j,r_{i,j}) \in \mathcal{V}} \text{SE}(\text{Testing}[i, j]) \right)^{1/2}$$

Simpan data($\text{iterasi}, \mathbf{W}, \mathbf{H}, \text{RMSE}(\text{iterasi}, \text{Testing}), \text{RMSE}(\text{iterasi}, \text{Training})$)

Iterasi 1 sd max iterasi

3.4. Ilustrasi dan Simulasi CF

Untuk memahami proses yang terjadi dalam algoritma 3-1, yang pertama kali dibutuhkan adalah data *rating*. Secara umum data tersebut berbentuk matriks, dimana baris-barisnya menyatakan indeks pengguna sedangkan kolom-kolomnya menyatakan indeks produk. Untuk lebih mudahnya, berikut disampaikan ilustrasi data *rating* film yang diperoleh dari dua puluh orang pengguna yang diminta untuk memberikan *rating* pada dua puluh dua judul film yang disediakan. Skala *rating* yang dipakai dalam ilustrasi ini menggunakan skala *likert* (1 sd 5). Deskripsi datanya dilihat pada Tabel 3-1 berikut.

Tabel 3-1: Deskripsi data rating

Data Ilustrasi (disimpan dalam file : u.data)		
No	Indikator	Angka
1	Jumlah Pengguna (N)	20
2	Jumlah Film (M)	22
3	Jumlah Data Rating	233
4	Jumlah Data Training	120
5	Jumlah Data Testing	113
6	Ukuran Matriks	20 x 22
7	Jumlah entri matriks	440
8	Prosentasi data rating	233/440=53.3%
9	<i>Sparsity</i>	46.7%

Untuk memproses data yang dideskripsikan dalam Tabel 3-1 ke dalam algoritma 3-1, data tersebut terlebih dulu dituliskan dalam notasi himpunan D berikut ini:

$$D = \{(q, (i, j), r_{ij}) | q \in Q, i \in U, j \in F, r_{ij} \in T\}$$

dimana:

q: indeks pengamatan ke-q,

i: indeks pengguna ke-i,

j: indeks film ke-j,

r_{ij} : rating yang diberikan oleh pengguna ke-i pada film ke-j.

$Q = \{1, 2, \dots, 233\}$ himpunan indeks pengamatan

$U = \{1, 2, \dots, 20\}$ himpunan indeks pengguna

$F = \{1, 2, \dots, 22\}$ himpunan indeks film

$T = \{1, 2, 3, 4, 5\}$ himpunan skor rating film.

Dilihat dari bentuk himpunan D maka penulisan skor *rating* film di atas dapat ditulis dengan notasi: $d_{(q)} = (q, (i, j, r_{ij}))$. Dalam himpunan tersebut, terdapat pemetaan q ke (i, j, r_{ij}) yang bersifat *surjective* yaitu pemetaan yang memiliki hubungan 1-1 dan pada.

Ilustrasi data yang terdapat pada Lampiran 1 menunjukkan bahwa untuk pengamatan yang ke 51, skor *rating* film yang diberikan oleh pengguna ke-13 pada film ke-11 adalah 3. Dengan menggunakan notasi himpunan, maka entri *rating* tersebut dapat ditulis dengan $d_{(51)} = (51, (13, 11, 3))$. Itulah deskripsi sederhana bagaimana suatu *rating* film diformulasikan dalam notasi himpunan D .

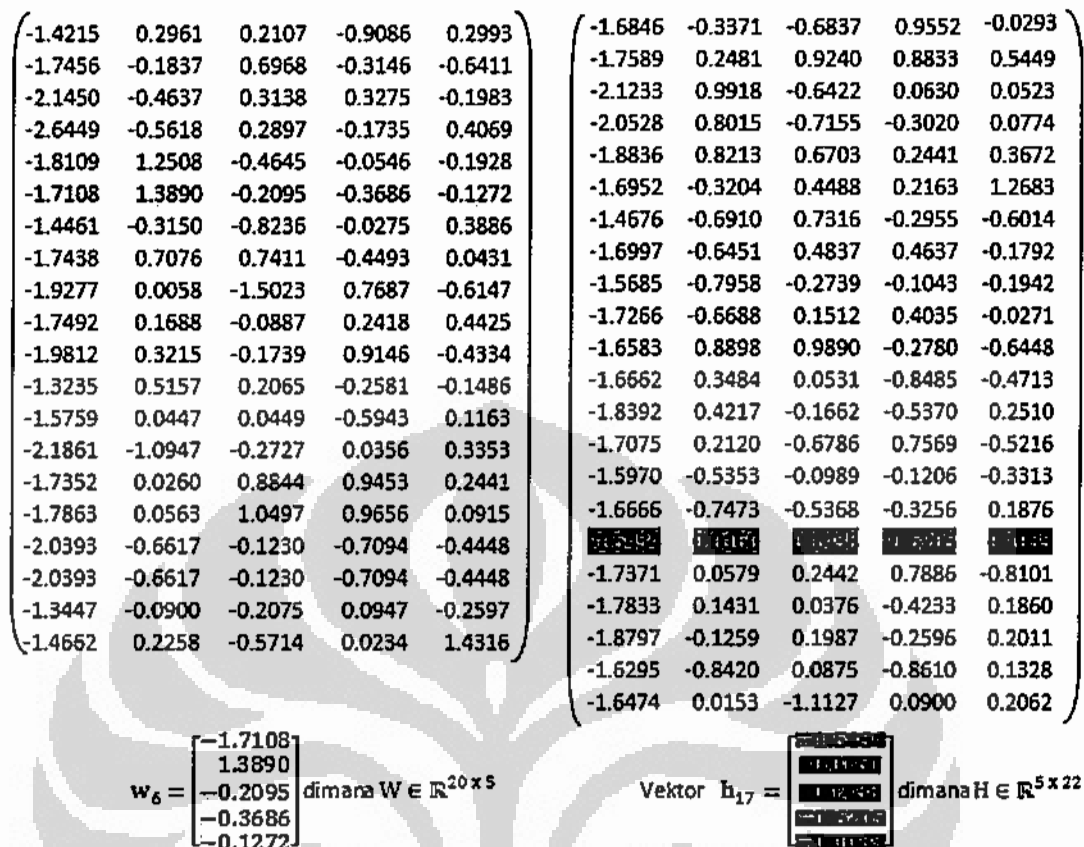
Sebelum data dimasukkan ke dalam program matlab, seharusnya data tersebut dipartisi terlebih dulu ke dalam dua kelompok data, yaitu himpunan data *training* (\mathcal{R}) dan himpunan data *testing* \mathcal{V} dimana kedua himpunan tersebut bersifat *disjoint* atau ditulis $\mathcal{R} \subset D, \mathcal{V} \subset D, \mathcal{R} \cup \mathcal{V} = D$ dan $\mathcal{R} \cap \mathcal{V} = \emptyset$.

$\forall (q, (i, j, r_{ij})) \in D, \exists (w_i, h_j)$ sedemikian sehingga $e_{i,j} = r_{i,j} - w_i^T h_j$, maka proses mencari error $e_{i,j}$ untuk setiap elemen $(q, (i, j, r_{ij})) \in D$, dapat dilakukan dengan terlebih dahulu menyusun $q, i, j, r_{ij}, w_i, h_j$ ke dalam Tabel 3-2 berikut ini:

Tabel 3-2 : Format susunan data dalam komputer

q, i, j dan r_{ij} diperoleh dari setiap elemen $(q, (i, j, r_{ij})) \in D$				vektor w_i dan h_j diperoleh dari matriks W dan H			error ke (i,j)
q	i	j	r_{ij}	w_i	h_j	$w_i^T h_j$	e_{ij}

Dengan memasukan data q, i, j dan r_{ij} kedalam algoritma 3.1 CF di subbab 3.3 diatas, maka matriks W dan H bisa ditentukan. Misalkan matriks W dan H optimal yang diperoleh dari simulasi tersebut adalah matriks berikut :



Gambar 3-2 Ilustrasi Matriks W dan H^T hasil eksperimen

Pada Gambar 3-2 vektor w_6 dan h_{17} hanya merupakan ilustrasi untuk membandingkan skor *rating* aktual dari data atau elemen himpunan ($q = 194, (i = 6, j = 17, r_{6,17} = 3)$) dengan prediksi *rating* untuk elemen yang bersesuaian. Berdasarkan perhitungan maka skor prediksinya adalah:

$\hat{r}_{6,17} = w_6^T h_{17} = 2.8481$. Maka besarnya kesalahan prediksi skor *rating* terhadap nilai aktual dari elemen ($q = 194, (i = 6, j = 17, r_{6,17} = 3)$) dapat dihitung sebagai selisih :

$e_{6,17} = r_{6,17} - w_6^T h_{17} = 3 - 2.8481 = 0.1519$. Sedangkan *squared error* untuk satu pengamatan tersebut adalah $(e_{6,17})^2 = 0.02307$. Bila semua titik pengamatan dianalisa nilai kesalahannya, maka besarnya kesalahan dari semua elemen tersebut akan tercermin dari besarnya RMSE. Semakin kecil nilai RMSE, maka kemampuan model dalam memprediksi akan semakin baik.

BAB 4 SIMULASI

Bab ini sebagian besar membahas simulasi model faktorisasi matriks probabilitas pada data real MovieLens yang terbagi ke dalam delapan subbab. Subbab pertama berisi tentang deskripsi data dan sumbernya. Subbab kedua menyajikan informasi spesifikasi komputer yang digunakan dalam penelitian, subbab ketiga desain eksperimen, subbab keempat inisialisasi proses simulasi, subbab kelima perhitungan akurasi model CF, subbab keenam menyajikan hasil eksperimen subbab ketujuh yaitu membentuk matriks prediksi dan diakhiri dengan subbab ke delapan yaitu interpretasi hasil penelitian.

4.1. Data

Data eksperimen diperoleh dari situs www.movielens.org yang dikelola oleh *GroupLens Research Project* Universitas Minnesota. Data ini terdiri dari : 100,000 skor *rating* yang berasal dari 943 pengguna pada 1682 produk film. Masing-masing pengguna memberikan *rating* film paling sedikit sedikit 20 judul film. Skor *rating* yang diberikan oleh pengguna menggunakan skala *likert* (1 sampai dengan 5) . Selain skor *rating* disajikan juga beberapa info demografik sederhana dari pengguna seperti umur, jenis kelamin dan pekerjaan.

Data ini sudah melalui proses pembersihan, artinya pengguna-pengguna yang merating film kurang dari 20 judul film dan tidak mengisi data demografik secara lengkap tidak akan dimasukkan dalam data penelitian. Data simulasi ini disimpan dalam file bernama *u.data* yang kemudian dipecah menjadi lima bagian dan membentuk 5 partisi yang saling lepas (*disjoint*) yaitu S1, S2, S3, S4, dan S5. Delapan puluh persen data digunakan sebagai data *training*, dan dua puluh persen sisanya digunakan sebagai data *testing*. Kelima partisi data tersebut susunannya diatur sedemikian secara bergantian untuk mendapatkan susunan yang berbeda-beda baik untuk data *training* maupun data *testing*.

Deskripsi data penelitian ini telah disusun dalam bentuk tabel seperti yang terlihat pada Tabel 4-1. Data rating film tersebut selanjutnya disimpan ke dalam suatu matriks yang berukuran 943 x 1682. Matriks inilah yang kemudian akan diproses dan dianalisa. Sedangkan bentuk partisi dari data penelitian ini dapat dilihat pada Tabel 4-2.

Tabel 4-1 Indikator Data Penelitian

Data MovieLens (disimpan dalam file : u.data)		
No	Indikator	Angka
1	Jumlah Pengguna (N)	943
2	Jumlah Produk (M)	1682
3	Jumlah Data Rating	100.000
4	Jumlah Data Training	80.000
5	Jumlah Data Testing	20.000
6	Ukuran Matriks	943 x 1682
7	Jumlah entri matriks	1.586.126
8	Prosentasi data rating	$100.000/1.586.126=6.3\%$
9	<i>Sparsity</i>	93.7%

Tabel 4-2 Komposisi Data *Training* dan Data *Testing* untuk setiap *Fold*

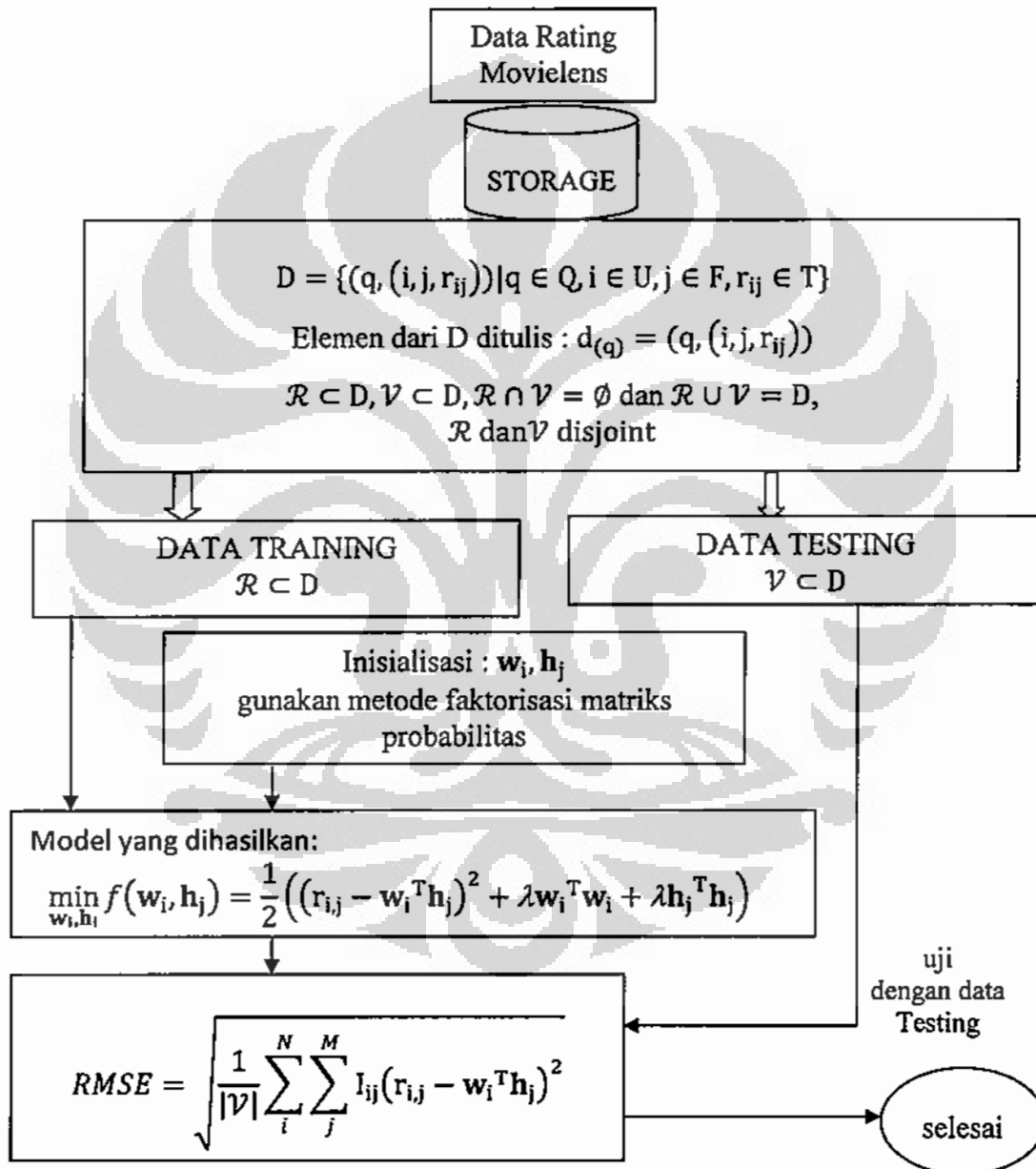
Data awal : nama file u.data					
S1	S2	S3	S4	S5	
Dipecah menjadi :					
<i>Fold 1</i>	nama file: u1.base				u1.test
	S1	S2	S3	S4	S5
<i>Fold 2</i>	nama file: u2.base				u2.test
	S1	S2	S3	S5	S4
<i>Fold 3</i>	nama file: u3.base				u3.test
	S1	S2	S4	S5	S3
<i>Fold 4</i>	nama file: u4.base				u4.test
	S1	S3	S4	S5	S2
<i>Fold 4</i>	nama file: u5.base				u5.test
	S2	S3	S4	S5	S1

4.2. Spesifikasi komputer

Semua proses simulasi dijalankan pada notebook IBM R51e 4TA Series dengan spesifikasi mesin sebagai berikut: Pentium Centrino 1.73 GHz ; Memori (RAM) : 750 MB dan HardDisk : 40 GB.

4.3. Desain Eksperimen

Desain ini dibuat untuk menentukan tahapan apa saja dan mulai dari mana proses pengolahan data ini dilakukan. Sesuai dengan tujuan semula, eksperimen ini bertujuan untuk menentukan nilai parameter model yang optimal yaitu: k (*range*) dan λ (*regularisasi*). Untuk mendapatkan nilai parameter tersebut dibuat suatu skema percobaan mulai dari masukan, proses, dan keluaran.



Gambar 4-1 Skema percobaan model faktorisasi matriks

Tetko, Livingstone, & Luik (1995) menyatakan : “jika RMSE pada data *testing* mulai meningkat, sementara RMSE pada training terus menurun, saat tersebut adalah kondisi dimana *overfitting* mulai terjadi. Ketika RMSE pada data *testing* mencapai minimum global, di sinilah model *fit* terjadi yaitu model yang memiliki prediksi dengan akurasi yang baik”.

Untuk melengkapi skema di atas, berikut adalah desain eksperimen untuk mendapatkan parameter model optimal k dan λ . Sedangkan parameter simulasi η (*learning rate*) dipakai untuk melihat tingkat ketelitian dalam eksperimen.

Tabel 4-3 Desain eksperimen penentuan nilai parameter model

Eksperimen ke-	EKSPERIMEN PENENTUAN PARAMETER MODEL					RMSE Minimum dicapai pada
	I	Parameter λ dan η (tetap)				
k berubah		k=1	k=2	...	Dst...	
RMSE						
<i>Fold 1</i>						
<i>Fold 2</i>						
<i>Fold 3</i>						
<i>Fold 4</i>						
	<i>Fold 5</i>					
Dari kelima <i>fold</i> tersebut cari k yang memberikan nilai RMSE terkecil.						
II	Parameter η dan k (tetap)					$\min_{\lambda} RMSE(testing)$ $\lambda = ?$
	λ berubah	λ_1	λ_2	...	Dst...	
	RMSE					
III	Parameter λ dan k (tetap)					$\min_{\eta} RMSE(testing)$ $\eta = ?$
	η berubah	η_1	η_1	...	Dst...	
	RMSE					

Dalam desain eksperimen pada Tabel 4-3, yang terlebih dahulu harus dilakukan adalah menentukan inisialisasi matriks fitur pengguna $W \in \mathbb{R}^{n \times k}$ dan matriks fitur produk $H \in \mathbb{R}^{k \times n}$, dimana n adalah banyaknya pengguna, m banyaknya produk dan k adalah jumlah fitur terpilih dalam faktorisasi matriks. Dengan desain eksperimen tersebut diharapkan hasilnya dapat menjawab masalah utama dalam penelitian ini yaitu mencari parameter model yang optimal.

Parameter model yang optimal adalah parameter memberikan keakuratan yang tinggi dalam memprediksi dan merekomendasi. Keakuratan model dapat terjadi bila RMSE yang dihasilkan oleh model mencapai nilai minimum global pada data *testing*. Dalam penelitian ini k dan λ merupakan parameter yang memiliki kontribusi dalam menentukan keakuratan.

Pemilihan k sangat berpengaruh terhadap besarnya ukuran matriks W dan H . Semakin besar nilai k akan semakin besar ukuran matriks W dan H . Akibatnya semakin besar nilai k maka waktu komputasi yang dicapai akan semakin lama. Besarnya peningkatan waktu operasi dapat dilihat dari *floating point operation* perkalian dua matriks $W_{(n \times k)}$ dan $H_{(k \times m)}$. Menurut Hunger (2007) *floating point operation* untuk perkalian dua matriks yang berukuran $n \times k$ dan $k \times m$ adalah :

$$\text{flops}(n, k, m) = n * m * (2 * k - 1)$$

Dengan memasukan $n=943$ dan $m=1682$, maka *floating point operation* untuk perkalian matriks W dan H adalah: $\text{flops}(943, k, 1682) = 943 * 1682 * (2 * k - 1)$. Dari persamaan tersebut terlihat bahwa penambahan satu point nilai k akan meningkatkan waktu komputasi sebesar $2 * n * m = 3.172.252$ satuan waktu operasi. Oleh karena itu k dipilih hanya 20 nilai, dari semua nilai k tersebut kemudian diambil satu nilai k yang memberikan RMSE terkecil pada data *testing*.

4.4. Inisialisasi Proses Simulasi

Simulasi diawali dengan memberikan inisialisasi dua matriks W dan H . Entri dari kedua matriks tersebut adalah bilangan acak yang diasumsikan berasal dari data yang berdistribusi normal multivariat berdimensi k . Pemilihan distribusi ini didasarkan pada asumsi yang disampaikan dalam tinjauan teoritis di bab tiga.

Dalam Matlab® perintah untuk membuat matriks inisialisasi tersebut cukup ditulis dengan perintah :

$W = \text{randn}(n, k)$ dan $H = \text{randn}(k, m)$

dimana W adalah matriks fitur pengguna, H matriks fitur film, n banyaknya pengguna aktif di *database*, m banyaknya film yang ditawarkan. Kedua matriks tersebut merupakan data masukan untuk program simulasi. Besarnya nilai n dan m

dapat diketahui dari sumber data yang digunakan, dalam eksperimen digunakan $n=943$ dan $m= 1682$.

4.5. Akurasi Model CF

Kualitas atau kinerja sistem rekomendasi tercermin dari besarnya nilai akurasi yang diperoleh dari penelitian. Terdapat beberapa jenis satuan ukur yang dapat digunakan untuk menentukan kualitas tersebut. Satuan ukur yang dipilih tersebut sangat bergantung pada metode CF yang diterapkan dalam membangun model (Su & Khoshgoftaar, 2009).

Menurut Herlocker, Konstan, Terveen, & Riedl (2004) secara umum kualitas kinerja sistem rekomendasi dapat diukur berdasarkan kategori *predictive accuracy metrics* yaitu dengan *Mean Absolute Error* dan RMSE. Pendapat yang sama juga disampaikan oleh Gunawardana & Shani (2009) bahwa berbagai satuan ukur yang sering digunakan dalam sistem rekomendasi adalah RMSE, yaitu metoda yang mengukur perbedaan antara preferensi prediksi rating dengan preferensi aktual dari produk-produk yang ditawarkan.

Berdasarkan hal tersebut maka RMSE dipilih sebagai satuan ukur yang akan digunakan untuk mengukur keakuratan model. Dari berbagai jurnal yang ditemui, banyak metode CF yang menggunakan satuan ukur ini. Penjelasan lebih lanjut satuan ukur RMSE untuk CF dapat dilihat pada uraian berikut ini.

Su & Khoshgoftaar, (2009) menyebutkan bahwa RMSE adalah satuan ukur yang digunakan untuk menilai keakuratan kinerja sistem rekomendasi. Secara matematik formulasi RMSE dinyatakan dengan:

$$RMSE = \sqrt{\frac{1}{|\mathcal{V}|} \sum_{(i,j) \in \mathcal{V}} (p_{i,j} - r_{i,j})^2}$$

Dimana :

$|\mathcal{V}|$: jumlah keseluruhan rating pada data testing

$p_{i,j}$: prediksi nilai rating yang diberikan oleh pengguna i pada produk j .

$r_{i,j}$: nilai rating sebenarnya yang diberikan oleh pengguna i pada produk j .

Dengan metoda *CF*, keakuratan atau kinerja suatu model dianggap baik bila memberikan nilai RMSE yang kecil (Su, 2008). RMSE akan mengukur perbedaan antara nilai rating prediksi yang dihasilkan model dengan rating aktual dari data testing. Semakin kecil nilai RMSE, maka model yang diperoleh semakin baik.

4.6. Hasil eksperimen

Subbab ini menyajikan beberapa temuan data yang diperoleh dari eksperimen. Tahapan dalam eksperimen tersebut mengikuti pola seperti yang tertera dalam Tabel 4-3 yaitu eksperimen ke I, ke II dan ke III. Pada eksperimen ke I difokuskan pada proses simulasi pencarian nilai k , kemudian diikuti dengan eksperimen ke II yaitu simulasi penentuan nilai λ . Selanjutnya diakhiri dengan eksperimen ke III yaitu melihat pengaruh η (*learning rate*) terhadap RMSE model faktorisasi matriks probabilitas.

4.6.1. Mencari nilai k

Eksperimen pencarian nilai k telah dicatat dalam Tabel 4-4. Konten dari tabel ini merupakan rangkuman hasil simulasi yang menyajikan kontribusi berbagai nilai k terhadap nilai RMSE pada masing-masing *fold*. Pada Tabel 4-4, untuk nilai k mulai dari $k=1$ sampai dengan $k=10$, RMSE terkecil diperoleh pada $k=4$ di *fold* ke 4 yaitu 0.9232. Sedangkan untuk $k=11$ sampai dengan $k=20$, RMSE terkecil diperoleh pada $k=17$ di *fold* ke 4 yaitu 0.9214.

Dari hasil eksperimen terhadap 20 nilai k pada masing-masing *fold* seperti yang tercantum dalam Tabel 4-4, RMSE minimum diperoleh pada $k=17$ pada *fold* ke 4. Pada kondisi nilai $k=17$ itulah model yang dihasilkan untuk sementara waktu telah mencapai kondisi yang optimal. Namun masih terdapat parameter lain yang harus diuji diantaranya λ yang sama-sama berpengaruh terhadap keoptimalan model.

Tabel 4-4 RMSE untuk berbagi nilai k mulai dari *Fold 1* sampai dengan *Fold 5*

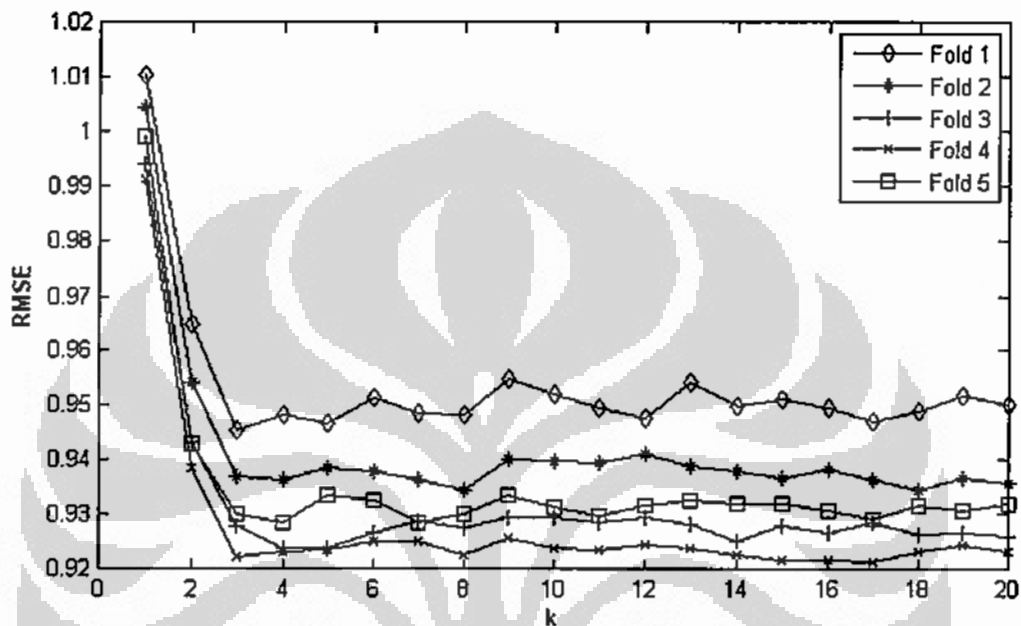
		k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Fold I	RMSE	1.0106	0.9649	0.9456	0.9481	0.9466	0.9515	0.9486	0.9484	0.955	0.9521
	Iterasi	432	351	304	272	186	182	126	127	152	118
II	RMSE	1.0046	0.9543	0.937	0.9364	0.9386	0.9381	0.9365	0.9345	0.9402	0.9398
	Iterasi	420	789	209	190	183	211	167	142	137	127
III	RMSE	0.9944	0.9433	0.9278	0.924	0.9239	0.9268	0.929	0.9277	0.9295	0.9296
	Iterasi	410	285	318	639	387	275	132	268	344	123
IV	RMSE	0.9914	0.9386	0.9224	0.9232	0.9236	0.9252	0.9251	0.9228	0.9258	0.9238
	Iterasi	265	309	271	222	157	125	106	137	132	121
V	RMSE	0.9992	0.943	0.9301	0.9285	0.9336	0.9327	0.9286	0.9302	0.9335	0.9315
	Iterasi	241	310	310	295	151	148	120	116	113	114
		k=11	k=12	k=13	k=14	k=15	k=16	k=17	k=18	k=19	k=20
I	RMSE	0.9494	0.9477	0.9543	0.9498	0.9511	0.9497	0.9469	0.9488	0.9517	0.9501
	Iterasi	115	113	121	114	109	115	113	110	116	108
II	RMSE	0.9394	0.9411	0.9388	0.938	0.9367	0.9382	0.9363	0.9345	0.9366	0.9359
	Iterasi	105	97	128	105	116	107	107	103	105	103
III	RMSE	0.9287	0.9295	0.9284	0.9251	0.9278	0.9268	0.9285	0.9264	0.9266	0.926
	Iterasi	123	107	113	105	111	111	104	104	105	117
IV	RMSE	0.9236	0.9246	0.9239	0.9228	0.9219	0.9218	0.9214	0.9232	0.9245	0.9234
	Iterasi	105	109	120	104	106	100	99	99	96	
V	RMSE	0.9298	0.9317	0.9326	0.9321	0.9319	0.9308	0.9293	0.9316	0.9309	0.9321
	Iterasi	99	106	108	103	102	101	102	97	97	93

Hasil simulasi yang tertera pada Tabel 4-4 diperoleh pada kondisi dimana nilai parameter $\lambda=0.2$, $\eta=0.00045$ dan nilai k mulai dari 1 sampai dengan 20. Berdasarkan nilai parameter tersebut maka nilai RMSE optimal yang dicapai model faktorisasi matriks probabilitas adalah 0.9214. Secara ringkas kondisi tersebut dapat dilihat pada Tabel 4-5

Tabel 4-5 Nilai RMSE terkecil dan kondisi parameter yang diperoleh dari Tabel 4-4

η	: 0.00045
λ	: 0.2
k	: 17 pada fold ke 4
Iterasi	: 99
RMSE	: 0.9214

Selain hasil eksperimen ke I dibuatkan dalam bentuk tabel, Gambar 4-2 menampilkan secara grafik pengaruh bervariasinya nilai k pada berbagai *fold*. Dari grafik tersebut dapat dilihat trend atau kecenderungan nilai RMSE. Juga dapat dilihat pada *fold* dan k mana saja dari hasil simulasi tersebut yang memberikan kontribusi dalam penentuan model yang optimal.



Gambar 4-2 Kurva RMSE untuk berbagai nilai k pada *fold* yang berbeda

Titik optimal pada Gambar 4-2 menunjukkan hal yang sama dengan data yang tertera di Tabel 4-5, yaitu RMSE terkecil adalah 0.9214. Nilai RMSE tersebut dicapai pada kondisi $\eta = 0.00045$, $\lambda = 0.2$, $k=17$, dan komposisi data pada *fold* ke 4. Walaupun demikian, namun keoptimalan model itu baru bersifat sementara saja. Artinya masih perlu dikaji seberapa jauh pengaruh parameter λ terhadap nilai RMSE. Oleh karena itu eksperimen ke II akan berlanjut ke tahap penentuan pengaruh nilai λ .

4.6.2. Mencari nilai λ

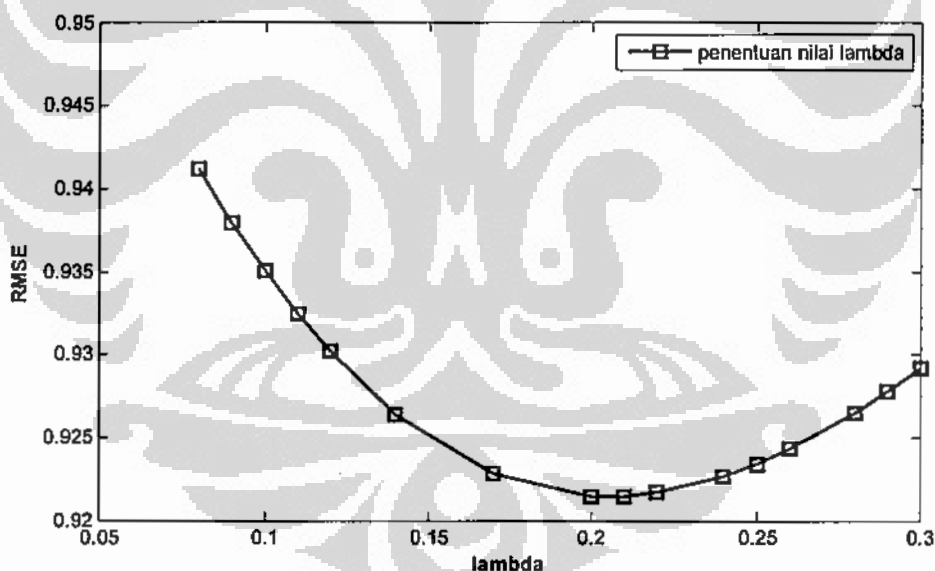
Dengan eksperimen yang dikontrol pada kondisi $\eta = 0.00045$ dan $k=17$ selanjutnya akan dianalisa sejauh mana pengaruh perubahan nilai λ terhadap nilai RMSE. Hasil eksperimen menunjukkan bahwa perubahan nilai RMSE dalam

pemodelan dapat dipengaruhi oleh besarnya nilai λ . Seberapa besar pengaruh perubahan tersebut dapat dilihat hasilnya pada Tabel 4-6 berikut:

Tabel 4-6 Nilai RMSE berdasarkan perubahan nilai parameter regularisasi λ

λ	0.08	0.09	0.1	0.11	0.12	0.14	0.17	0.2
RMSE	0.9411	0.9379	0.9350	0.9325	0.9302	0.9264	0.9228	0.9214
Iterasi	57	58	62	65	69	76	88	99
λ	0.21	0.22	0.24	0.25	0.26	0.28	0.29	0.3
RMSE	0.9215	0.9217	0.9227	0.9234	0.9243	0.9265	0.9278	0.9292
iterasi	109	115	130	138	148	168	183	198

Pada eksperimen ke II, perubahan nilai RMSE yang diakibatkan oleh bervariasinya nilai λ dapat diplot kedalam bentuk kurva seperti yang tertera pada Gambar 4-3. Kurva yang muncul menyerupai bentuk parabola, dimana RMSE minimum bersesuaian dengan nilai minimum dari kurva tersebut.



Gambar 4-3 Kurva RMSE berdasarkan pemilihan nilai lambda (λ)

Dari Gambar 4-3 terlihat bahwa RMSE terkecil terjadi pada saat pemilihan nilai parameter $\lambda = 0.2$. Pada kondisi tersebut RMSE yang dicapai adalah 0.9214. Dari kedua eksperimen tersebut dapat ditarik kesimpulan bahwa model optimal dicapai terjadi saat parameter $k=17$ dan $\lambda = 0.2$.

4.6.3. Menentukan pengaruh *learning rate* η terhadap nilai RMSE

Walaupun eksperimen ke I dan ke II sebenarnya eksperimen yang sudah memenuhi syarat untuk menentukan parameter model yang optimal, namun untuk mengetahui tingkat ketelitian hasil yang diperoleh, perlu dilakukan langkah lebih lanjut. Langkah tersebut terdapat pada pemilihan nilai parameter simulasi η . Oleh karena itu dalam eksperimen ke III akan disimulasikan seberapa jauh pengaruh η terhadap RMSE. Dari percobaan yang telah dilakukan, hasilnya menunjukkan pola seperti yang tertera pada Tabel 4-7 berikut :

Tabel 4-7 Nilai RMSE berdasarkan perubahan η

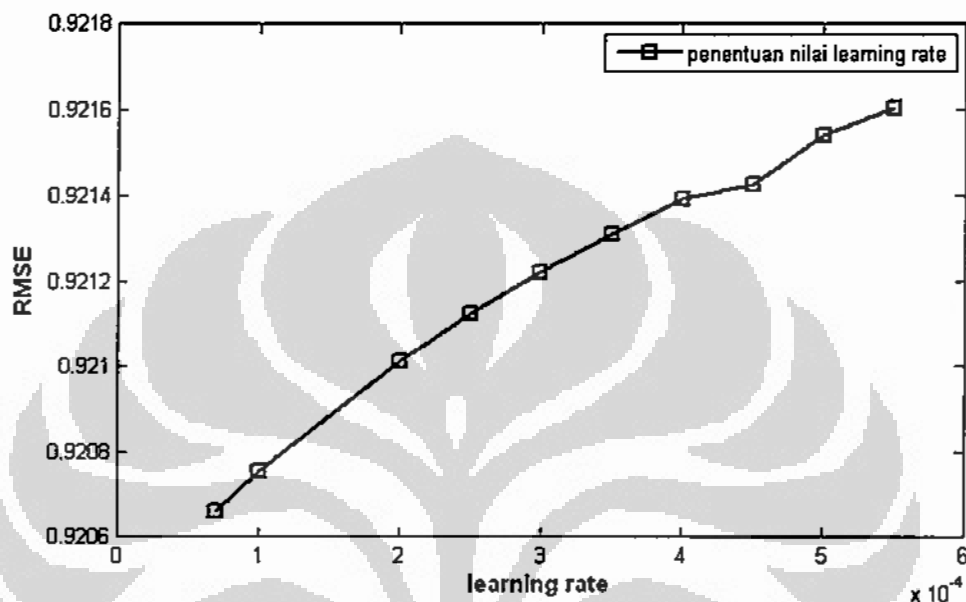
Learning rate η	0.00007	0.0001	0.0002	0.00025	0.0003
RMSE	0.920661	0.920753	0.921013	0.921122	0.921221
Iterasi	608	428	220	178	150
Learning rate η	0.00035	0.0004	0.00045	0.0005	0.00055
RMSE	0.921311	0.921394	0.921425	0.921541	0.921606
Iterasi	130	115	99	94	86

Pada Tabel 4-7 di atas, ternyata semakin kecil nilai η yang dipilih maka semakin banyak pula jumlah iterasi yang diperlukan. Akibatnya waktu komputasi yang diperlukan untuk mencapai konvergen akan semakin besar. Oleh karena itu jika tidak ada perbedaan nilai RMSE untuk 3 digit pertama di belakang decimal, sebaiknya carilah nilai η sedemikian sehingga titik konvergensi dicapai pada jumlah iterasi minimum.

Pada Gambar 4-4 diperlihatkan kurva pengaruh learning rate η terhadap nilai RMSE. Semakin kecil η , RMSE yang dihasilkan akan semakin kecil dan konvergen di nilai 0.920. Learning rate (η) bukanlah parameter model penelitian. Hal ini tercermin dari definisi fungsi *loss* yang dihasilkan. Dalam fungsi *loss*, hanya k dan λ yang merupakan parameter model.

Dengan telah dilaluinya ketiga tahapan eksperimen di atas, maka dapat diketahui seberapa besar pengaruh setiap parameter tersebut dalam pemodelan. Dari sini bisa juga lihat seberapa baikkah model yang dibuat dapat memprediksi dan merekomendasikan kebutuhan pengguna. Secara teoretis bila RMSE yang dihasilkan oleh model mendekati nol, maka model yang terbentuk mendekati

model yang ideal. Namun hingga saat ini belum ditemui referensi yang menunjukkan keakuratan model sebaik itu. Dengan nilai RMSE=0.920 model yang terbentuk sudah bisa dikatakan baik. Hal ini didasari bahwa beberapa jurnal terbitan terbaru memiliki kisaran RMSE di sekitar 0.9.

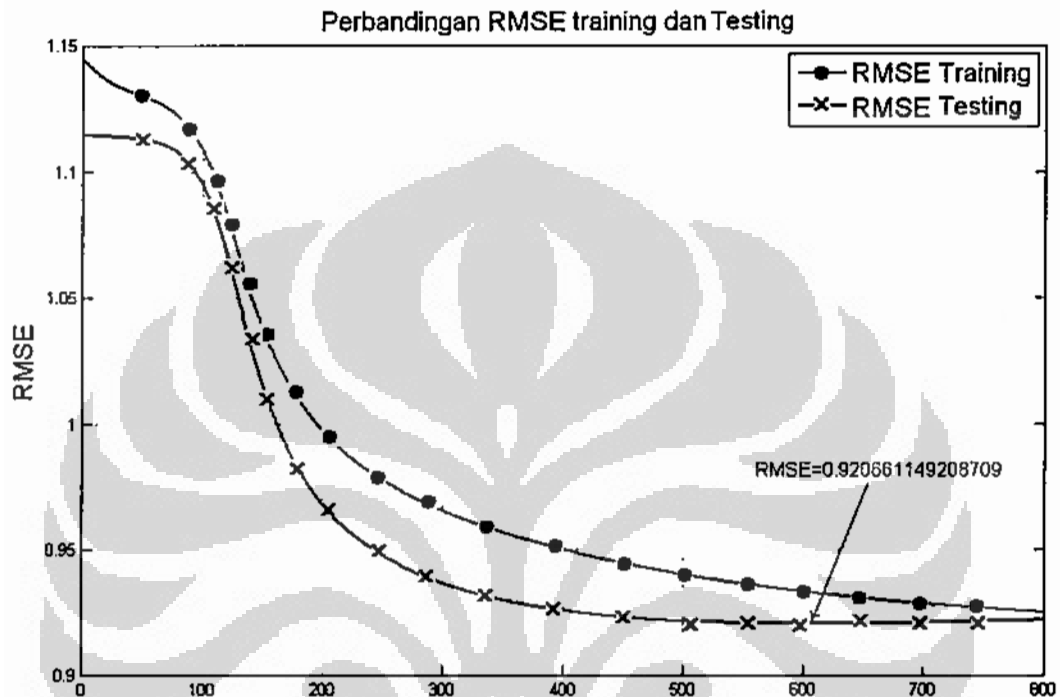


Gambar 4-4 Kurva RMSE terhadap berbagai η

Walaupun ketiga tahapan eksperimen dalam menentukan RMSE terkecil sudah diproses dengan algoritma yang sama, namun dalam penentuan matriks W dan H disyaratkan bahwa parameter masukannya sudah merupakan parameter yang sudah final. Artinya tidak terdapat lagi proses coba-coba untuk mencari nilai parameter model yang optimal. Dengan parameter inputnya yang sudah optimal, maka akurasi terbaik dari model yang dihasilkan bisa dianalisa berdasarkan keluaran program simulasi. Gambar 4-5 merupakan salah satu contoh keluaran program simulasi.

Untuk mengetahui matriks W dan H yang optimal dari Gambar 4-5, acuan yang digunakan untuk membacanya merujuk pada pendapat Tetko, Livingstone, & Luik (1995) yang menyatakan : “jika RMSE pada data *testing* mulai meningkat, sementara RMSE pada training terus menurun, saat tersebut adalah kondisi dimana *overfitting* mulai terjadi. Ketika RMSE pada data *testing* mencapai minimum global, pada posisi inilah model *fit* terjadi yaitu model yang

memiliki prediksi dengan akurasi yang baik”. Berdasarkan acuan tersebut walaupun pada setiap iterasi dalam Gambar 4-5 menghasilkan matriks W dan H , namun posisi W dan H optimal hanya dapat diperoleh saat iterasi mencapai nilai RMSE minimum global pada data testing.



Gambar 4-5 Perbandingan RMSE Data Training dan Data Testing dalam setiap Iterasi

Dalam Gambar 4-5, kondisi optimal dicapai pada saat nilai iterasi=608. Pada kondisi tersebut semua informasi yang terkandung dalam model baik W , H , k , λ , η , error training dan error testing dapat disimpan secara permanen dengan perintah matlab :

```
Save('datafinal.mat','W','H','K','eta','lambda','aa_m','aa_p','rating','err_valid','err_train')
```

Matriks W dan H yang tersimpan di *database* inilah yang selanjutnya digunakan untuk memprediksi dan merekomendasikan produk-produk mana saja yang sesuai dengan selera konsumen.

4.7. Matriks Prediksi

Matriks prediksi adalah matriks yang dituliskan dengan notasi \hat{R} . Matriks \hat{R} diperoleh dari perkalian matriks W dan matriks H atau ditulis $\hat{R} = WH$. Matriks W dan H yang dipilih adalah matriks yang memenuhi optimasi seperti yang dinyatakan dalam persamaan 3.28. Berdasarkan eksperimen, nilai optimal model dicapai pada kondisi nilai parameter $k = 17, \lambda = 0.2$ dan $\eta = 0.00007$.

Dengan diperolehnya matriks \hat{R} , maka skor *rating* entri ke (i,j) dari matriks R yang sebelumnya tidak dirating bisa diprediksi. Entri ke (i,j) data awal dari matriks R tetap disimpan dan penyelesaiannya lebih fokus pada nilai prediksi entri ke (i,j) dari matriks R yang tidak dirating. Hal ini tentu saja dilakukan agar entri ke (i,j) matriks R yang ada skor ratingnya tidak ditimpa oleh entri ke (i,j) dari matriks \hat{R} . Cara sederhana yang bisa dilakukan adalah dengan menentukan matriks indikator dari matriks R (matriks rating film data MovieLens). Misalkan matriks indikator dari matriks R adalah matriks bobot $B \in \mathbb{R}^{N \times M}$, dimana matriks B adalah matriks yang entrinya didefinisikan dengan:

$$b_{i,j} = \begin{cases} 1, & \text{Jika entri matrik } r_{i,j} = 0 \\ 0, & \text{Jika entri matrik } r_{i,j} \neq 0 \end{cases}$$

Maka matriks R dapat diperbaharui menjadi matriks R baru yang dapat dirumuskan dengan :

$$R^{\text{update}} \leftarrow R + B \odot \hat{R}$$

4.8. Interpretasi hasil penelitian

Model faktorisasi matriks probabilitas pada data MovieLens mencapai titik optimal saat nilai parameter : $k = 17, \lambda = 0.2$ dan learning rate = 0.00007. Pada kondisi tersebut nilai RMSE yang dicapai oleh model adalah 0.920661. Dalam faktorisasi matriks, parameter model k berkaitan dengan banyaknya fitur atau variabel laten dari data *rating*. Fitur atau variabel laten dari data rating film biasa disebut genre. Sehingga nilai $k = 17$ dapat diartikan sebagai banyaknya genre film dalam data MovieLens.

Jumlah genre dari model faktorisasi matriks ini memang sedikit berbeda dengan yang disampaikan oleh *GroupLens Research Project* Universitas Minnesota yang menghasilkan 18 genre yaitu : *Action*|1, *Adventure*|2, *Animation*|3, *Children's*|4, *Comedy*|5, *Crime*|6, *Documentary*|7, *Drama*|8, *Fantasy*|9, *Film-Noir*|10, *Horror*|11, *Musical*|12, *Mystery*|13, *Romance*|14, *Sci-Fi*|15, *Thriller*|16, *War*|17, dan *Western*|18. Perbedaan jumlah genre tersebut diduga ada dua genre dari 18 genre di atas yang mengklaster menjadi satu genre.

RMSE = 0.920661 dicapai pada iterasi ke 608 . Dengan jumlah iterasi yang sedemikian besar, tentu saja merupakan pemborosan dari segi waktu komputasi. Akan lebih efektif bila iterasi yang dicapai jauh lebih kecil dari 608 namun RMSE yang dihasilkan tetap sama untuk tiga digit dibelakang desimal. Cara yang bisa dilakukan untuk mengatasi masalah tersebut adalah dengan merubah nilai parameter simulasi η (*learning rate*) dari 0.00007 menjadi 0.0001. Dengan perubahan tersebut, jumlah iterasi yang diperlukan untuk mencapai titik optimal turun drastis dari 608 menjadi 428 akan tetapi nilai RMSE tetap sama dengan sebelumnya untuk tiga digit pertama (lihat Tabel 4-7).

Sedangkan $\lambda=0.2$ menunjukkan bahwa besarnya koefisien regularisasi untuk model faktorisasi matriks adalah 0.2. Koefisien ini berkaitan dengan fungsi *loss* dari setiap data pengamatan yaitu:

$$f(\mathbf{w}_i, \mathbf{h}_j) = \frac{1}{2} \left((r_{i,j} - \mathbf{w}_i^T \mathbf{h}_j)^2 + \lambda \mathbf{w}_i^T \mathbf{w}_i + \lambda \mathbf{h}_j^T \mathbf{h}_j \right)$$

BAB 5 PENUTUP

5.1. Kesimpulan

Kesimpulan yang diperoleh dari hasil penelitian ini adalah :

1. Nilai parameter yang optimal untuk model faktorisasi matriks probabilitas pada data MovieLens diperoleh saat nilai : $k=17$, $\lambda=0.2$, learning rate=0.00007. Pada kondisi tersebut kinerja atau akurasi model menunjukkan akurasi yang paling baik. Matriks W dan H yang diperoleh dari eksperimen dicapai saat nilai RMSE = 0.920661.
2. Pada nilai RMSE tersebut matriks prediksi $\hat{R} = WH$ dapat diterapkan untuk memprediksi entri yang masih kosong pada matriks sparse MovieLens.
3. Fungsi *loss* dari model faktorisasi matriks probabilitas direpresentasikan dalam bentuk persamaan :

$$f(W, H) = \frac{1}{2} \left(\|X \odot (R - WH)\|_F^2 + 0.2(\|W\|_F^2 + \|H\|_F^2) \right)$$

dimana :

W adalah matriks yang berukuran 943×17

H adalah matriks yang berukuran 17×1682

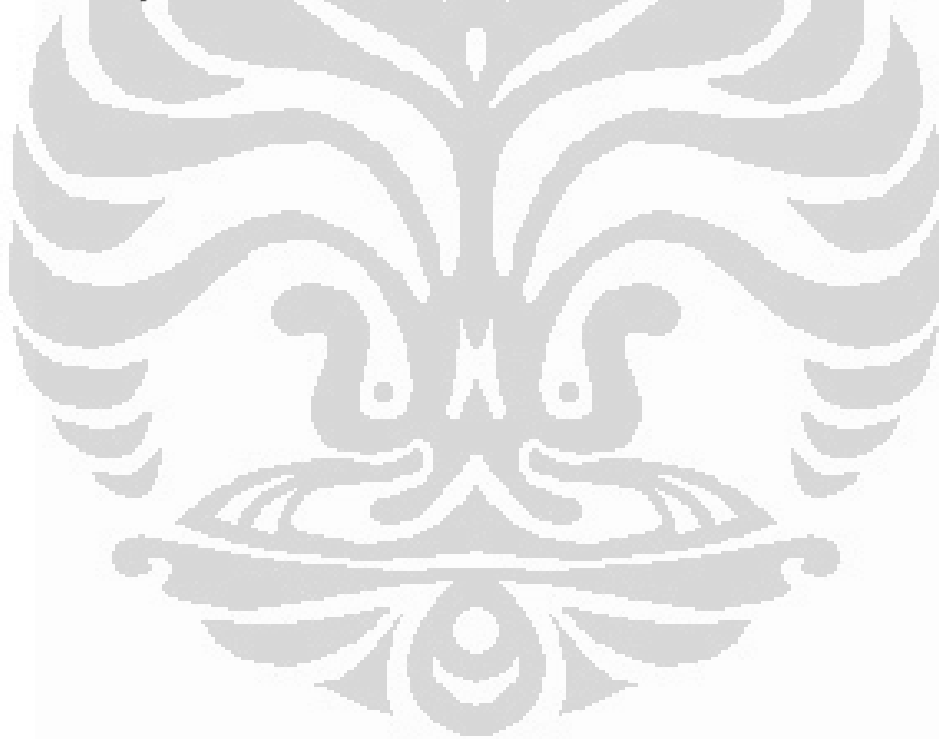
R adalah matriks skor rating yang berukuran 943×1682

X adalah matriks indikator yang berukuran 943×1682

5.2. Saran

- 1 Nilai k terbaik yang diperoleh dari eksperimen dicapai pada saat nilai $k=17$, Nilai tersebut hampir mencerminkan banyaknya *genre* film dalam data MovieLens yang jumlahnya 18. Perbedaan jumlah *genre* ini kemungkinan disebabkan oleh masih adanya kesalahan dalam mengambil asumsi-asumsi yang dipakai dalam penelitian. Oleh karena itu perbaikan model masih diperlukan.

- 2 Dalam matriks kovarian dalam persamaan (3.9) telah diasumsikan bahwa antara variabel satu dengan variabel lainnya tidak berkorelasi. Hal ini dilakukan hanya untuk mempermudah analisis. Namun akan lebih baik bila variabel tersebut diasumsikan saling berkorelasi yang dalam realitanya bisa saja demikian.
- 3 Diawal sudah disampaikan bahwa skor rating suatu produk yang diberikan oleh seorang pengguna tidak dipengaruhi oleh pengguna lainnya, artinya skor rating berlaku independen. Akan tetapi apa yang terjadi di dunia real biasanya terdapat kasus-kasus dimana pemilihan produk oleh seorang pengguna bisa dipengaruhi oleh pengguna lainnya yang memiliki *trust* (rasa percaya). Oleh karena itu akan lebih baik bila *Social Network Analysis* diikutsertakan dalam Faktorisasi Matriks Probabilitas.



DAFTAR PUSTAKA

- Aldrich, J. (1997). R.A. Fisher and Making of Maximum Likelihood 1912-1922. *Statistical Sciences* , 12 (3), 162-176.
- Alpaydin, E. (2004). *Introduction to Machine Learning*. London: The MIT Press Cambridge.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer.
- Caro-Lopera, F. J., Leiva, V., & Balakrishnan, N. (2012). Connection between the Hadamard and matrix products with an application to matrix-variate Birnbaum-Saunders distributions. *Journal of Multivariate Analysis* , 126-139.
- DasGupta, A. (2011). *Probability for Statistics and Machine Learning Fundamental and Advanced Topics*. Springer.
- Deshpande, M., & Karypis, G. (2004). Item-Based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems* , 22 (1), 143-177.
- Dueck, D., & Frey, B. J. (2004). *Probabilistic Sparse Matrix Factorization*. Technical Report, Dept of Computer Science, Toronto.
- Ercan, E. (2010). *Probabilistic Matrix Factorization Based Collaborative Filtering with Implicit Trust Derived from Review ratings information*. Middle East Technical University.
- Gong, S., Ye, H., & Tan, H. (2009). Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System. *Pacific-Asia Conference on Circuit, Communications and System* (pp. 690-693). IEEE Computer Society Washington, DC, USA.
- Graus, M. (2011). *Understanding the latent Features of Matrix factorization Algorithms in Movie Recommender Systems*. Eindhoven: Eindhoven University of Technology.
- Gunawardana, A., & Shani, G. (2009). A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research* , 2935-2962.
- Hao, M. (2009). *Learning to Recommend*. Chinese University, Computer Science. Hongkong: UNI Dissertation Publishing.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* , 22 (1), 5-53.

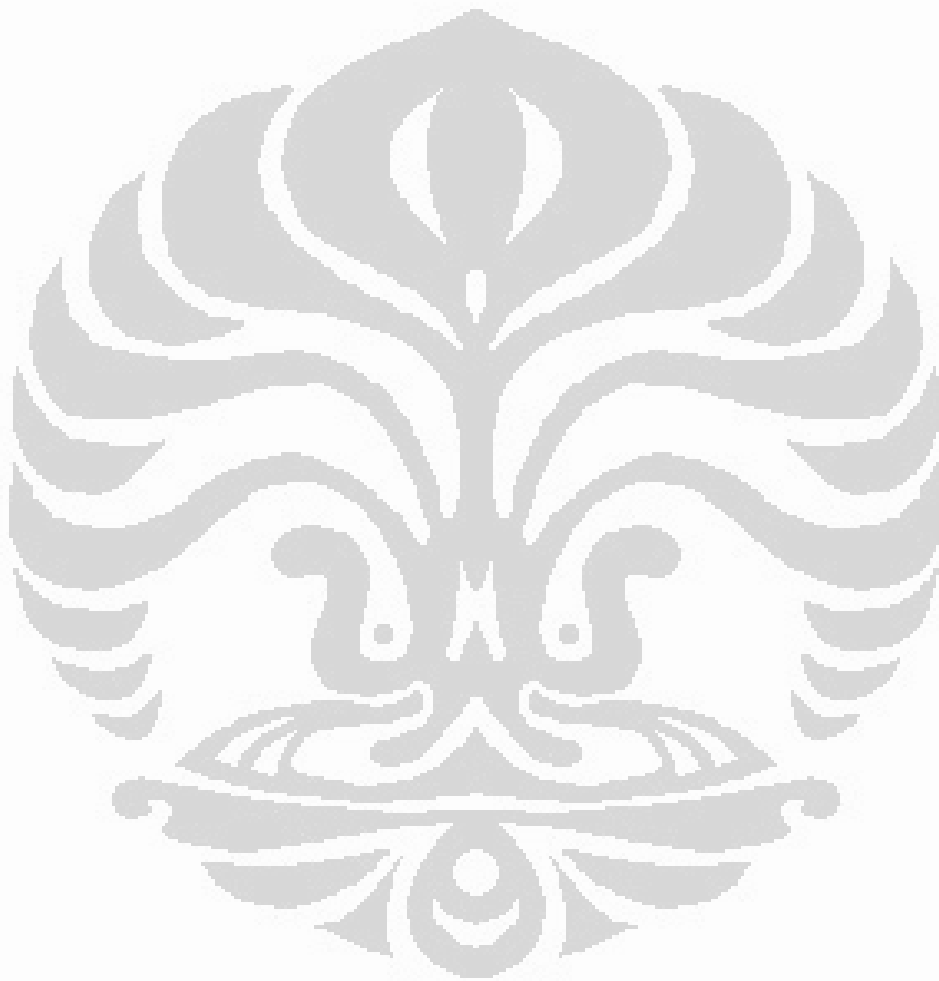
- Hofmann, T. (2004). Latent Semantic Model for Collaborative Filtering. *ACM Transactions on Information Systems* , 22 (1), 89-115.
- Hunger, R. (2007). *Floating Point Operations in Matrix-Vector Calculus (version 1.3)*. Munchen: Technische Universitat Munchen.
- Koch, K. (2007). *Introduction to Bayesian Statistics* (2nd ed.). Verlag Berlin Heidelberg: Springer.
- Lakshminarayanan, B., Bouchard, G., & Cedric, A. (2011). Robust Bayesian Matrix Factorization. *Proceedings of 14th Int. Conf. on Artificial Intelligence and Statistics*, 15, pp. 425-433. Fort Lauderdale, FL, USA.
- Lee, D. D., & Seung, H. S. (2000). Algorithms for Non-negative Matrix Factorization. *Advances in Neural Information Processing Systems 13 Proceedings of the 2000 Conference* (pp. 556-562). MIT Press.
- Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., et al. (2009). A Brief History of the Internet. *ACM SIGCOMM Computer Communication Review* , 39 (5), 22-31.
- Lim, Y. J., & Teh, Y. W. (2007). Variational Bayesian Approach to Movie Rating Prediction. *ACM 978-1-59593-834-3 KDD* , (pp. 15-21). San Jose California USA.
- Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations Item-to-Item Collaborative Filtering. *IEEE Internet Computing* , 76-80.
- Liu, T. Y. (2011). *Learning to Rank for Information Retrieval* (1st ed.). Springer.
- Mcginty, L., & Smyth, B. (2006). Adaptive Selection: An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems. *International Journal of Electronic Commerce archive* , 11 (2), 35-57.
- Myung, I. J. (2003). Tutorial on maximum likelihood estimation. (Elsevier, Ed.) *Journal of Mathematical Psychology* , 90-100.
- Pilaszky, I. (2009). *Factorization-Based large Scale Recommendation Algorithms*. PhD. Thesis Budapest University of Technology and Economics, Budapest Hungary.
- Rencher, A. C. (2002). *Methods of Multivariate Analysis* (2nd ed.). John Wiley & Son, Inc.
- Rennie, J. D., & Srebro, N. (2005). Fast Maximum Margin Matrix Factorization for Collaborative Prediction. *Proceeding of 22nd International Conference on Machine Learning*, (pp. 713-719). Bonn, Germany.
- Roman, S. (2008). *Advanced Linear Algebra, Graduate Texts in Mathematics* (3rd ed.). Springer.

- Saad, Y. (2003). *Iterative methods for sparse linear systems* (2nd ed.). The Society for Industrial Mathematics.
- Salakhutdinov, R., & Mnih, A. (2008). Probabilistic Matrix Factorization. *Advances in Neural Information Processing Systems 20* (pp. 1257-1264). Cambridge, MA, USA: MIT Press.
- Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. T. (2000). *Application of Dimensionality Reduction in Recommender System - A Case Study*. In ACM WebKDD 2000 Web Mining for E-Commerce Workshop.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommender system algorithms. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web*, (pp. 285-295). New York, NY USA.
- Shani, G., Heckerman, D., & Brafman, R. I. (2005). An MDP-Based Recommender System. *Journal of Machine Learning Research*, 1265-1295.
- Shaqlaih, A. S. (2010). *Model Selection Using Information Theory Approach*. Oklahoma: University of Oklahoma.
- Shi, Y., Larson, M., & Hanjalic, A. (2010). List-wise Learning to Rank with Matrix Factorization for Collaborative Filtering. *Proceeding of the fourth ACM conference on Recommender systems*, (pp. 269-272). New York.
- Strunjas, S. (2008). *Algorithms and Models for Collaborative Filtering from Large Information Corpora*. Ph.D. Thesis, University of Belgrade, Belgrade, Serbia.
- Su, X. (2008). *Collaborative Filtering Using Machine Learning and Statistical Techniques*. Dissertation, Florida Atlantic University, Boca Raton FL.
- Su, X., & Khoshgoftaar, T. M. (2009). *Review Article A survey of Collaborative Filtering Techniques*. Hindawi Publishing Corporation.
- Tetko, I. V., Livingstone, D. J., & Luik, A. I. (1995). Neural Network Studies. I Comparison of Overfitting and Overtraining. *J. Chem. Inf. Comput. Sci*, 35, 826-833.
- Walpole, R. E. (1992). *Pengantar Statistika* (3 ed.). Jakarta: PT. Gramedia Pustaka Utama.
- Wang, B., Tao, Z., & Hu, J. (2010). Improving the Diversity of User-based Top-N Recommendation by Cloud Model. *The 5th International Conference on Computer Science & Education*, (pp. 1323-1327). Hefei, China.
- Wang, J., de Vries, A. P., & Reinders, M. J. (2006). Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion.

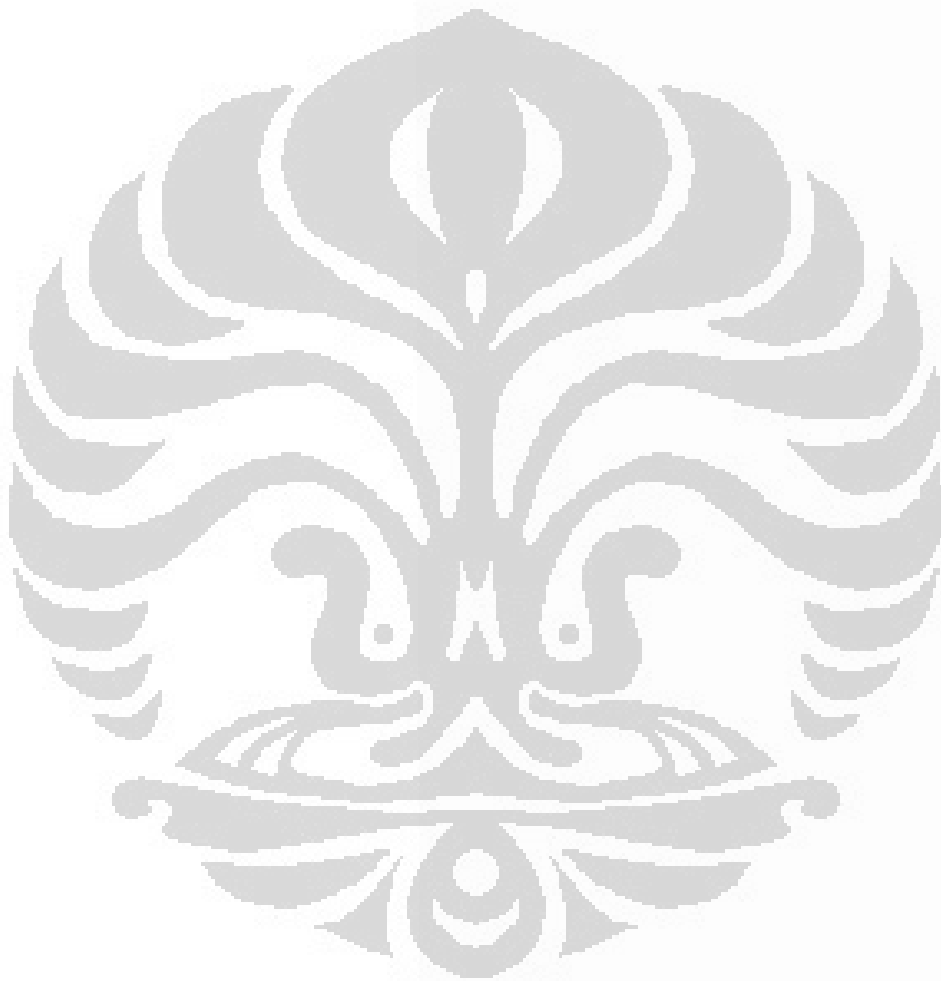
Wen, Z. (2008, December 12). Recommendation System Based on Collaborative Filtering.

Zhang, S., Wang, W., Ford, J., & Makedon, F. (2005). Using Singular Value Decomposition Approximation for Collaborative Filtering. *Proceedings of the Seventh IEEE International Conference on E-Commerce Technology* (pp. 1-8). IEEE Computer Society.

Zhao, Z. D., & Shang, M. S. (2010). User-based Collaborative Filtering Recommendation Algorithms on Hadoop. *Third International Conference on Knowledge and Data Mining* (pp. 478-481). IEEE Computer Society.



LAMPIRAN



Lampiran 1 Ilustrasi Data rating

$D = \{(q, (i, j, r_{ij})) | q \in Q, i \in U, j \in F, r_{ij} \in T\}$, elemen dari D terdiri dari 233

rating film dari 20 pengguna dan 22 judul film dengan skor *rating* 1 sd 5. dimana elemennya adalah:

$$\{$$

(1, (15, 20, 3)),	(26, (2, 14, 2)),	(51, (13, 11, 3)),
(2, (7, 7, 2)),	(27, (20, 11, 1)),	(52, (1, 22, 2)),
(3, (18, 8, 4)),	(28, (4, 19, 5)),	(53, (9, 2, 3)),
(4, (15, 7, 3)),	(29, (10, 4, 3)),	(54, (10, 10, 4)),
(5, (3, 15, 3)),	(30, (12, 14, 3)),	(55, (8, 8, 3)),
(6, (16, 22, 2)),	(31, (17, 14, 3)),	(56, (12, 18, 2)),
(7, (19, 12, 2)),	(32, (16, 8, 4)),	(57, (17, 13, 4)),
(8, (17, 8, 4)),	(33, (8, 3, 4)),	(58, (15, 19, 3)),
(9, (8, 13, 4)),	(34, (14, 11, 2)),	(59, (7, 20, 1)),
(10, (16, 10, 4)),	(35, (20, 18, 1)),	(60, (19, 11, 2)),
(11, (1, 11, 3)),	(36, (15, 6, 4)),	(61, (5, 11, 4)),
(12, (15, 13, 3)),	(37, (14, 8, 4)),	(62, (8, 9, 2)),
(13, (3, 6, 4)),	(38, (17, 6, 3)),	(63, (2, 9, 3)),
(14, (14, 10, 4)),	(39, (16, 20, 3)),	(64, (11, 4, 3)),
(15, (9, 1, 5)),	(40, (16, 3, 3)),	(65, (3, 10, 4)),
(16, (5, 2, 3)),	(41, (10, 7, 2)),	(66, (5, 12, 3)),
(17, (11, 9, 3)),	(42, (15, 10, 3)),	(67, (10, 2, 3)),
(18, (15, 2, 5)),	(43, (9, 5, 2)),	(68, (13, 6, 3)),
(19, (3, 22, 3)),	(44, (2, 16, 3)),	(69, (9, 15, 4)),
(20, (3, 4, 4)),	(45, (7, 15, 2)),	(70, (19, 3, 3)),
(21, (2, 13, 3)),	(46, (9, 13, 3)),	(71, (10, 19, 3)),
(22, (5, 19, 4)),	(47, (18, 20, 4)),	(72, (15, 15, 3)),
(23, (20, 1, 3)),	(48, (4, 18, 4)),	(73, (15, 3, 3)),
(24, (11, 22, 3)),	(49, (11, 17, 3)),	(74, (17, 4, 4)),
(25, (10, 8, 3)),	(50, (10, 3, 5)),	(75, (2, 8, 3)),
(76, (19, 22, 2)),	(101, (11, 19, 3)),	(126, (12, 1, 1)),
(77, (8, 15, 2)),	(102, (16, 5, 4)),	(127, (6, 4, 5)),
(78, (4, 4, 5)),	(103, (3, 20, 4)),	(128, (7, 8, 2)),
(79, (6, 1, 2)),	(104, (15, 5, 4)),	(129, (12, 20, 3)),
(80, (9, 3, 5)),	(105, (17, 12, 4)),	(130, (1, 8, 1)),
(81, (5, 17, 3)),	(106, (15, 9, 2)),	(131, (9, 20, 3)),
(82, (20, 21, 2)),	(107, (5, 6, 2)),	(132, (5, 1, 3)),
(83, (17, 21, 4)),	(108, (1, 5, 3)),	(133, (8, 12, 4)),
(84, (8, 19, 3)),	(109, (6, 19, 3)),	(134, (15, 16, 2)),
(85, (3, 18, 4)),	(110, (10, 18, 2)),	(135, (4, 7, 4)),
(86, (2, 20, 3)),	(111, (10, 13, 3)),	(136, (19, 21, 3)),
(87, (14, 5, 3)),	(112, (17, 15, 4)),	(137, (10, 6, 3)),
(88, (20, 8, 2)),	(113, (6, 11, 4)),	(138, (13, 15, 3)),
(89, (12, 17, 2)),	(114, (14, 13, 4)),	(139, (12, 19, 3)),
(90, (1, 6, 3)),	(115, (8, 18, 3)),	(140, (9, 17, 2)),
(91, (14, 9, 4)),	(116, (17, 7, 4)),	(141, (20, 20, 3)),
(92, (8, 2, 4)),	(117, (11, 7, 2)),	(142, (19, 6, 2)),
(93, (7, 21, 3)),	(118, (9, 7, 2)),	(143, (1, 10, 2)),
(94, (10, 1, 4)),	(119, (4, 1, 4)),	(144, (12, 5, 3)),
(95, (3, 5, 4)),	(120, (12, 3, 3)),	(145, (18, 15, 4)),
(96, (15, 22, 2)),	(121, (5, 8, 2)),	(146, (7, 12, 2)),

$$\}$$

Lanjutan lampiran 1

(97, (19, 9, 2)), (122, (8,17, 3)), (147, (18,16, 4)),
 (98, (19,14, 2)), (123, (8,16, 2)), (148, (14,21, 5)),
 (99, (7,18, 2)), (124, (17,11, 3)), (149, (9,22, 5)),
 (100, (4,12, 4)), (125, (12,12, 2)), (150, (9, 6, 2)),

 (151, (14, 1, 4)), (176, (6,14, 3)), (201, (6,18, 3)),
 (152, (10,12, 3)), (177, (16,21, 2)), (202, (18, 3, 4)),
 (153, (19, 2, 2)), (178, (14, 6, 4)), (203, (2,22, 2)),
 (154, (11, 6, 3)), (179, (6, 7, 1)), (204, (13,16, 3)),
 (155, (8,14, 2)), (180, (13,13, 3)), (205, (20,15, 2)),
 (156, (10,21, 2)), (181, (7,17, 2)), (206, (11, 2, 3)),
 (157, (12, 7, 2)), (182, (4,22, 4)), (207, (4, 8, 5)),
 (158, (8, 7, 3)), (183, (9,12, 3)), (208, (13,14, 2)),
 (159, (14,22, 5)), (184, (2,19, 3)), (209, (1,14, 2)),
 (160, (13,21, 3)), (185, (9,14, 5)), (210, (4, 2, 5)),
 (161, (1,20, 3)), (186, (20, 9, 2)), (211, (11,13, 3)),
 (162, (17,18, 3)), (187, (15, 8, 4)), (212, (5, 7, 2)),
 (163, (14,20, 5)), (188, (13, 7, 2)), (213, (18, 2, 2)),
 (164, (18,18, 3)), (189, (2, 1, 3)), (214, (12, 2, 2)),
 (165, (1,12, 3)), (190, (19, 8, 3)), (215, (17, 2, 2)),
 (166, (14,19, 4)), (191, (6,12, 4)), (216, (13, 2, 2)),
 (167, (2,18, 3)), (192, (7, 4, 4)), (217, (5,10, 2)),
 (168, (11, 1, 4)), (193, (20, 7, 1)), (218, (5,18, 3)),
 (169, (4,15, 4)), (194, (6,17, 3)), (219, (9,16, 4)),
 (170, (7,11, 1)), (195, (6,10, 2)), (220, (8,20, 3)),
 (171, (7, 1, 3)), (196, (4, 6, 5)), (221, (6,15, 2)),
 (172, (1,19, 3)), (197, (13, 1, 2)), (222, (16,16, 2)),
 (173, (15,12, 2)), (198, (12,16, 1)), (223, (1, 7, 2)),
 (174, (7, 3, 3)), (199, (18, 5, 3)), (224, (12,11, 3)),
 (175, (3,21, 4)), (200, (7, 9, 3)), (225, (18,19, 4)),

 (226, (17,16, 4)),
 (227, (1,17, 3)),
 (228, (17, 5, 3)),
 (229, (6, 6, 2)),
 (230, (7, 5, 2)),
 (231, (12, 6, 2)),
 (232, (15,11, 3)),
 (233, (19,10, 2)), }

Lampiran 2 Elemen himpunan \mathcal{R} dan \mathcal{V}

Elemen himpunan $\mathcal{R} \subset D$: (data training)

(143, 1, 10, 2))	(131, 9, 20, 3))	(71, 10, 19, 3))
(169, 4, 15, 4))	(1, 15, 20, 3))	(86, 2, 20, 3))
(90, 1, 6, 3))	(63, 2, 9, 3))	(100, 4, 12, 4))
(233, 19, 10, 2))	(182, 4, 22, 4))	(224, 12, 11, 3))
(226, 17, 16, 4))	(94, 10, 1, 4))	(55, 8, 8, 3))
(177, 16, 21, 2))	(119, 4, 1, 4))	(120, 12, 3, 3))
(187, 15, 8, 4))	(184, 2, 19, 3))	(30, 12, 14, 3))
(79, 6, 1, 2))	(204, 13, 16, 3))	(203, 2, 22, 2))
(180, 13, 13, 3))	(200, 7, 9, 3))	(92, 8, 2, 4))
(212, 5, 7, 2))	(19, 3, 22, 3))	(13, 3, 6, 4))
(134, 15, 16, 2))	(174, 7, 3, 3))	(96, 15, 22, 2))
(216, 13, 2, 2))	(104, 15, 5, 4))	(109, 6, 19, 3))
(54, 10, 10, 4))	(51, 13, 11, 3))	(89, 12, 17, 2))
(206, 11, 2, 3))	(150, 9, 6, 2))	(82, 20, 21, 2))
(159, 14, 22, 5))	(21, 2, 13, 3))	(194, 6, 17, 3))
(126, 12, 1, 1))	(81, 5, 17, 3))	(56, 12, 18, 2))
(9, 8, 13, 4))	(3, 18, 8, 4))	(24, 11, 22, 3))
(66, 5, 12, 3))	(157, 12, 7, 2))	(25, 10, 8, 3))
(207, 4, 8, 5))	(33, 8, 3, 4))	(231, 12, 6, 2))
(52, 1, 22, 2))	(78, 4, 4, 5))	(43, 9, 5, 2))
(53, 9, 2, 3))	(38, 17, 6, 3))	(148, 14, 21, 5))
(127, 6, 4, 5))	(5, 3, 15, 3))	(172, 1, 19, 3))
(40, 16, 3, 3))	(93, 7, 21, 3))	(70, 19, 3, 3))
(22, 5, 19, 4))	(199, 18, 5, 3))	(181, 7, 17, 2))
(170, 7, 11, 1))	(189, 2, 1, 3))	(196, 4, 6, 5))
(183, 9, 12, 3))	(8, 17, 8, 4))	(132, 5, 1, 3))
(195, 6, 10, 2))	(229, 6, 6, 2))	(130, 1, 8, 1))
(225, 18, 19, 4))	(228, 17, 5, 3))	(151, 14, 1, 4))
(128, 7, 8, 2))	(39, 16, 20, 3))	(192, 7, 4, 4))
(64, 11, 4, 3))	(124, 17, 11, 3))	(222, 16, 16, 2))
(208, 13, 14, 2))	(107, 5, 6, 2))	(115, 8, 18, 3))
(185, 9, 14, 5))	(91, 14, 9, 4))	(74, 17, 4, 4))
(133, 8, 12, 4))	(230, 7, 5, 2))	(49, 11, 17, 3))
(76, 19, 22, 2))	(75, 2, 8, 3))	(67, 10, 2, 3))
(118, 9, 7, 2))	(28, 4, 19, 5))	(147, 18, 16, 4))
(209, 1, 14, 2))	(84, 8, 19, 3))	(68, 13, 6, 3))
(112, 17, 15, 4))	(179, 6, 7, 1))	(125, 12, 12, 2))
(223, 1, 7, 2))	(57, 17, 13, 4))	(193, 20, 7, 1))
(7, 19, 12, 2))	(103, 3, 20, 4))	(190, 19, 8, 3))
(186, 20, 9, 2))	(98, 19, 14, 2))	(36, 15, 6, 4))

Elemen himpunan $\subset D$: (data testing)

(47, 18,20, 4))	(16, 5, 2, 3))	(135, 4, 7, 4))
(45, 7,15, 2))	(11, 1, 11, 3))	(197, 13, 1, 2))
(12, 15,13, 3))	(87, 14, 5, 3))	(220, 8,20, 3))
(175, 3,21, 4))	(83, 17, 21, 4))	(217, 5,10, 2))
(136, 19,21, 3))	(191, 6, 12, 4))	(69, 9,15, 4))
(201, 6,18, 3))	(65, 3, 10, 4))	(113, 6,11, 4))
(211, 11,13, 3))	(138, 13, 15, 3))	(171, 7, 1, 3))
(153, 19, 2, 2))	(88, 20, 8, 2))	(144, 12, 5, 3))
(10, 16,10, 4))	(20, 3, 4, 4))	(37, 14, 8, 4))
(168, 11, 1, 4))	(48, 4, 18, 4))	(198, 12,16, 1))
(166, 14,19, 4))	(214, 12, 2, 2))	(141, 20,20, 3))
(35, 20,18, 1))	(162, 17, 18, 3))	(116, 17, 7, 4))
(139, 12,19, 3))	(85, 3, 18, 4))	(97, 19, 9, 2))
(156, 10,21, 2))	(215, 17, 2, 2))	(163, 14,20, 5))
(77, 8,15, 2))	(2, 7, 7, 2))	(152, 10,12, 3))
(117, 11, 7, 2))	(155, 8, 14, 2))	(4, 15, 7, 3))
(23, 20, 1, 3))	(73, 15, 3, 3))	(167, 2,18, 3))
(145, 18,15, 4))	(121, 5, 8, 2))	(137, 10, 6, 3))
(34, 14,11, 2))	(80, 9, 3, 5))	(213, 18, 2, 2))
(178, 14, 6, 4))	(102, 16, 5, 4))	(61, 5,11, 4))
(210, 4, 2, 5))	(58, 15, 19, 3))	(218, 5,18, 3))
(160, 13,21, 3))	(6, 16, 22, 2))	(227, 1,17, 3))
(59, 7,20, 1))	(111, 10, 13, 3))	(41, 10, 7, 2))
(176, 6,14, 3))	(165, 1, 12, 3))	(18, 15, 2, 5))
(188, 13, 7, 2))	(99, 7, 18, 2))	(205, 20,15, 2))
(17, 11, 9, 3))	(27, 20, 11, 1))	(106, 15, 9, 2))
(44, 2,16, 3))	(60, 19, 11, 2))	(62, 8, 9, 2))
(202, 18, 3, 4))	(221, 6, 15, 2))	(164, 18,18, 3))
(14, 14,10, 4))	(123, 8, 16, 2))	(219, 9,16, 4))
(42, 15,10, 3))	(105, 17, 12, 4))	(142, 19, 6, 2))
(46, 9,13, 3))	(149, 9, 22, 5))	(114, 14,13, 4))
(15, 9, 1, 5))	(101, 11, 19, 3))	(146, 7,12, 2))
(232, 15,11, 3))	(140, 9, 17, 2))	(50, 10, 3, 5))
(161, 1,20, 3))	(95, 3, 5, 4))	
(26, 2,14, 2))	(122, 8, 17, 3))	
(29, 10, 4, 3))	(32, 16, 8, 4))	
(158, 8, 7, 3))	(110, 10, 18, 2))	
(108, 1, 5, 3))	(31, 17, 14, 3))	
(72, 15,15, 3))	(173, 15, 12, 2))	
(129, 12,20, 3))	(154, 11, 6, 3))	