

**LAPORAN**  
**SISTEM REKOMENDER MOVIE**  
**MENGGUNAKAN BAHASA PEMROGRAMAN “R”**



**Disusun Oleh :**

**Niko Fitrianto      16.01.63.0028**

**UNIVERSITAS STIKUBANK SEMARANG (UNISBANK)**

**FALKUTAS TEKNOLOGI INFORMASI**

**TEKNIK INFORMATIKA**

**2017**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Deskripsi Permasalahan**

Di tengah banyaknya informasi yang senantiasa membanjiri kita setiap detik, kita pun seakan menjadi kehilangan kendali atau kehilangan pegangan tentang informasi apa yang kita inginkan. Sehingga kemudian lahirlah sistem rekomendasi untuk membantu kita menemukan hal-hal yang kita sukai, informasi-informasi yang kita butuhkan dan semacamnya. Perkembangan teknologi telah memberikan dampak yang positif dalam kehidupan masyarakat. Karena sebagian besar masyarakat semakin merasakan informasi sebagai salah satu kebutuhan pokok. Kecepatan dan keakuratan dalam memperoleh informasi yang dibutuhkan telah mendorong transformasi masyarakat tradisional menjadi masyarakat informasi sehingga adanya keterikatan antara konsumen dengan penyedia barang maupun jasa semakin tinggi.

Sistem rekomendasi juga pada umumnya ditujukan untuk individu yang kekurangan pengalaman atau kompetensi yang cukup untuk mengevaluasi banyaknya jumlah alternatif item yang ada pada suatu kasus tertentu. Contohnya saya yang baru-baru ini menyukai musik beraliran *dubstep*, saya baru tau satu lagu *dubstep* yang bagus dan ingin mendengarkan lagu-lagu *dubstep* bagus yang lainnya. Karena baru mendengarkan satu lagu, bisa dibilang pengalaman saya dalam musik *dubstep* sangatlah minim. Karena itulah saya lantas mencari rekomendasi tentang musik-musik *dubstep* lainnya yang bagus di situs-situs seperti *Youtube*, *LastFM*, atau *SoundCloud*.

Sistem rekomendasi telah banyak digunakan oleh hampir sebagian besar bisnis area dimana konsumen perlu membuat suatu keputusan atau rekomendasi pilihan dari informasi yang disediakan (Sharda, N., 2010). Area pariwisata merupakan salah satu contoh bisnis area yang menerapkan sistem rekomendasi untuk membantu para wisatawan dalam membuat keputusan bagi perjalanan mereka. Internet dan world wide web menyediakan banyak informasi dibidang pariwisata karena pariwisata memiliki pengalaman menarik dan sensual bagi para wisatawan (Sharda, N., 2010) namun sangat sulit untuk menemukan informasi wisata yang sesuai dengan keinginan para penggunanya. Maka dibuatlah suatu sistem rekomendasi sebagai alat signifikan

bagi industri pariwisata atau perjalanan untuk menawarkan serta merekomendasikan tempat-tempat wisata kepada para wisatawan yang sesuai dengan keinginan mereka (Zins, A.H *et al.*, 2004).

Terdapat beberapa metode yang digunakan untuk mendukung cara kerja sistem rekomendasi dalam menghasilkan sebuah informasi diantaranya seperti *demographic recommendation*, *collaborative recommendation*, *content-based recommendation* dan *knowledge based recommendation*. Metode *collaborative* dan *content-based* adalah metode yang sering digunakan pada sistem rekomendasi karena teknik ini menyaring informasi berdasarkan keinginan pengguna dan berdasarkan content yang disediakan (Sebastia, L *et al.*, 2009). Namun terdapat beberapa kekurangan dalam kedua metode tersebut. Kekurangan pada metode *content-based* yaitu ketika seorang pengguna meminta rekomendasi suatu item yang memiliki jenis konten yang berbeda dengan item-item yang pernah dipilih oleh pengguna sebelumnya. Kekurangan pada metode *collaborative* terdapat pada saat suatu item yang dimasukkan dan sama sekali belum ada yang memberi rating, maka item tersebut tidak dapat direkomendasikan ke pengguna manapun.

Maka untuk menutupi kelemahan pada metode-metode tersebut dapat dilakukan dengan menggunakan metode *hybrid* yaitu menggabungkan beberapa metode yang terdapat pada sistem rekomendasi (Burke, R., 2007) untuk menghasilkan item rekomendasi yang sesuai dengan keinginan pengguna. Untuk mendukung cara kerja metode tersebut maka diperlukan algoritma *classification* yang akan mendukung kemampuan sistem rekomendasi dalam memberikan informasi yang sesuai. Salah satunya adalah algoritma *Nearest Neighbor* (NN).

Pada *recommender system* kualitas dari rekomendasi yang dihasilkan menjadi sebuah hal yang penting untuk diperhatikan agar *user* puas dengan hasil rekomendasi yang diberikan. Oleh karena itu perlu dicari solusi untuk permasalahan ini.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas maka rumusan masalahnya adalah bagaimana mengolah data menggunakan *movies recommender system* dengan *recommenderlab* beserta implementasinya menggunakan bahasa “R”.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Kajian Deduktif

##### 2.1.1 Sistem Rekomendasi

Sistem rekomendasi merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna (Ungkawa, et al., 2013). Penerapan rekomendasi didalam sebuah sistem biasanya melakukan prediksi suatu item, seperti rekomendasi film, musik, buku, berita dan lain sebagainya yang menarik user. Sistem ini berjalan dengan mengumpulkan data dari user secara langsung maupun tidak (Fadlil & Mahmudy, 2010).

Pengumpulan data secara langsung dapat dilakukan sebagai berikut : 1. Meminta user untuk melakukan rating pada sebuah item. 2. Meminta user untuk melakukan ranking pada item favorit setidaknya memilih satu item favorit. 3. Memberikan beberapa pilihan item pada user dan memintanya memilih yang terbaik. 4. Meminta user untuk mendaftar item yang paling disukai atau item yang tidak disukainya. Pengumpulan data dengan tidak langsung berhubungan dengan seorang user, dilakukan dengan cara mengamati item yang dilihat oleh seorang user pada sebuah *web ecommerce*. Dari data hasil yang dikumpulkan tersebut, kemudian diolah dengan menggunakan algoritma tertentu. Setelah itu, hasilnya tersebut dikembalikan lagi kepada user sebagai sebuah rekomendasi item dengan parameter dari user tersebut. Sistem rekomendasi juga merupakan salah satu alternatif sebagai mesin pencari suatu item yang dicari oleh user. Dalam pembangunan sistem rekomendasi, ada beberapa macam metode untuk menyelesaikan permasalahan, antara lain *user-based collaborative filtering*, *content-based filtering*, dan *hybrid*. Namun beberapa peneliti menambahkan metode baru, yaitu *knowledge based recommendation*. Secara umum, Metode *user-based collaborative filtering* menggunakan *feedback*, ulasan dan rating untuk mendapatkan hasil rekomendasi. Metode *content-based* memberikan rekomendasi dengan membangun profile pengguna. Metode *hybrid-based* menggabungkan dua atau lebih metode. Penggabungan dilakukan dengan tujuan saling melengkapi kurang dari metode yang digunakan. Sedangkan metode *knowledge-based* menggunakan pola pengetahuan untuk memberikan hasil rekomendasi.

### 2.1.2 Content-Based Filtering

Ada berbagai macam metode pendekatan yang digunakan untuk menyelesaikan permasalahan pada sistem rekomendasi, antara lain user-based collaborative filtering, contentbased filtering dan hybrid. Dalam kasus pemilihan pekerjaan untuk mahasiswa program sudi Teknik Informatika UAJY, metode yang digunakan adalah metode content-based filtering. *Content-based filtering* memberikan suatu rekomendasi berdasarkan hasil analisa kemiripan item yang telah dinilai oleh penggunanya (Adi, 2010) *Content-Based Filtering* membentuk profil penggunanya berdasarkan atribut pembentuk suatu item. *Cosine Similarity* yang tertinggi yang akan menjadi hasil rekomendasi.

*Content-Based Filtering* membentuk profil penggunanya berdasarkan atribut pembentuk suatu item. Sebagai contoh untuk suatu dokumen, atribut pembentuknya adalah katakata yang terdapat pada dokumen tersebut. Parameter pembentuk profil pengguna ini juga diberi nilai bobot berdasarkan kriteria tertentu. Adapun langkah – langkah algoritmanya :

1. Suatu item barang dibagi-bagi berdasarkan suatu vektor komponen pembentuknya.
2. Sistem akan membuat profil pengguna berdasarkan bobot vektor komponen pembentuk suatu item. Pembuatan profil pengguna dapat menggunakan algoritma TF-IDF (*term frequency-invers document frequency*). TF adalah jumlah term dalam suatu dokumen. Sedangkan nilai IDF dapat dihitung menggunakan rumus:

$$idf_i = \log \left( \frac{n}{df_i} \right)$$

n merupakan jumlah semua dokumen sedangkan df adalah jumlah dokumen yang memiliki term i.

3. Berdasarkan profil pengguna tersebut, sistem akan memperkirakan penilaian suka atau tidak suka suatu item berdasarkan analisis kemiripan profil pengguna dengan vektor komponen pembentuk item. Jika sistem memperkirakan bahwa item tersebut akan disukai oleh pengguna maka item tersebut akan direkomendasikan ke pengguna.

### 2.1.3 Collaborative Filtering

Ide utama dalam sistem rekomendasi collaborative filtering adalah untuk memanfaatkan opini pengguna lain yang ada untuk memprediksi item yang mungkin

akan disukai/diminati oleh seorang pengguna (Ricci et al, 2011). Kualitas rekomendasi yang diberikan dengan menggunakan metode ini sangat bergantung dari opini pengguna lain (neighbor) terhadap suatu item. Belakangan diketahui bahwa melakukan reduksi neighbor (yaitu dengan memotong neighbor sehingga hanya beberapa pengguna yang memiliki kesamaan / similarity tertinggi sajalah yang akan digunakan dalam perhitungan) mampu meningkatkan kualitas rekomendasi yang diberikan (Adomavicious dan Kwon, 2007). Sistem rekomendasi collaborative filtering beroperasi di dalam sebuah ruang dua dimensi User x Item. Rating yang dapat diberikan oleh seorang pengguna terhadap sebuah item dapat direpresentasikan sebagai  $R$  (bilangan bulat tidak negatif atau bilangan real dengan jarak tertentu), dan sistem rekomendasi mencoba untuk memprediksi rating yang akan diberikan seorang user untuk sebuah item yang belum pernah ia beri rating sebelumnya. Misalnya, terdapat lima pengguna  $u_1, u_2, u_3, \dots, u_5$  dan lima item  $i_1, i_2, i_3, \dots, i_5$ . Sebuah sistem rekomendasi ingin memprediksi berapa rating yang akan diberikan oleh  $u_1$  kepada  $i_5$ .

	Item $i_1$	Item $i_2$	Item $i_3$	Item $i_4$	Item $i_5$
User $u_1$	5	7	5	7	?
User $u_2$	5	7	5	7	9
User $u_3$	5	7	5	7	9
User $u_4$	6	6	6	6	5
User $u_5$	6	6	6	6	5

Tabel 2.1 Matrix User x Item pada Sistem Rekomendasi Collaborative Filtering  
Terdapat sebuah cara yang sering digunakan dalam menghitung / memprediksi rating pengguna  $u$  terhadap item  $i$  dengan mengkalkulasikan rata-rata rating pengguna lain (neighbor) (Adomavicious dan Kwon, 2007), yaitu rumus weighted sum:

$$R'(u, i) = \frac{1}{\sum_{u' \in N(u)} |sim(u, u')|} \sum_{u' \in N(u)} sim(u, u') * R(u', i)$$

Dimana:

- $R'(u, i)$  adalah prediksi rating pengguna  $u$  terhadap item  $i$ .
- $R(u', i)$  adalah rating pengguna  $u'$  (pengguna selain  $u$ ) terhadap item  $i$ .
- $sim(u, u')$  adalah kemiripan antara pengguna  $u$  dengan  $u'$ .
- $N(u)$  adalah satu set pengguna lain yang mirip dengan pengguna  $u$ .

### 2.1.4 Hybrid

Masing-masing teknik sistem rekomendasi memiliki kelebihan dan kelemahannya tersendiri. Karenanya, sistem rekomendasi hybrid yang menggabungkan kekuatan dari model dan algoritma yang berbeda untuk mengatasi masalah-masalah yang telah disebutkan di atas menjadi target penelitian yang sedang marak dikembangkan (Jannach et al, 2011). Terdapat beberapa cara penggabungan yang dapat dilakukan dalam metode hybrid yaitu :

#### 2.1.4.1 Penggabungan Linear (*Linear Combination*)

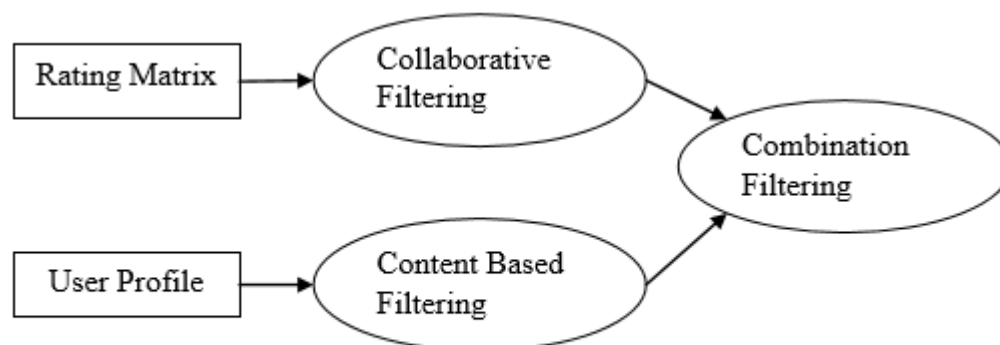
Penggabungan ini menggabungkan hasil perhitungan beberapa teknik rekomendasi secara linear. Penggabungan ini menghitung prediksi rating secara terpisah terlebih dahulu, kemudian hasil dari masing-masing metode tersebut digabungkan menjadi satu. Claypool et al (1999) menggunakan rumus weighted average untuk menggabungkan hasil tersebut.

$$R_{hybrid} = (w_1R_1 + w_2R_2 + ..w_nR_n)/(w_1 + w_2 + ..w_n)$$

Dimana :

- $R_{hybrid}$  merupakan prediksi rating hybrid.
- $w_n$  merupakan bobot prediksi rating dengan teknik/metode ke-n.
- $R_n$  merupakan prediksi rating dengan teknik / metode ke-n.

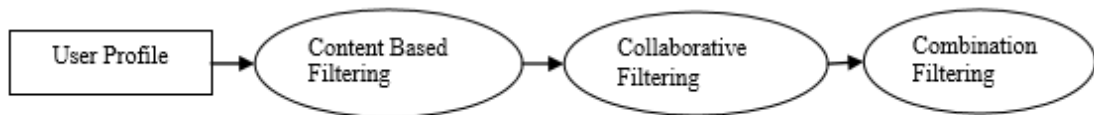
Gambar 2.1 merupakan contoh penggambaran dari penggabungan secara linear antara teknik *collaborative filtering* dengan *content based*.



**Gambar 2.1** Linear Combination

#### 2.1.4.2 Penggabungan secara Sekuensial (*Sequential Combination*)

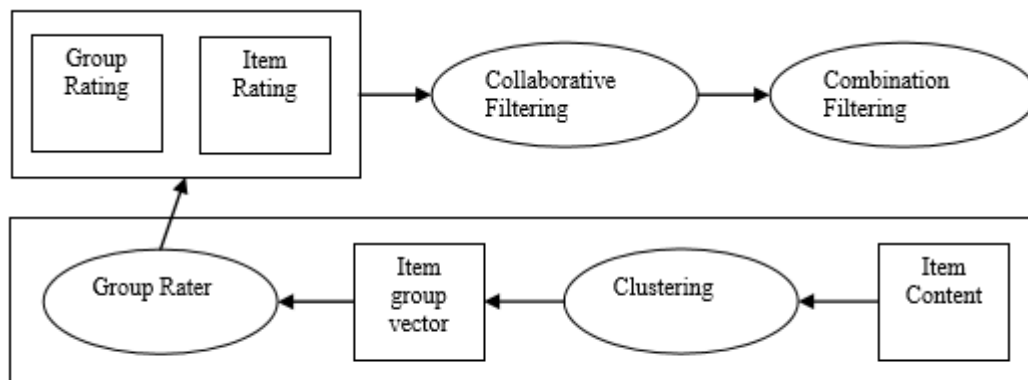
Penggabungan ini melakukan perhitungan pada salah satu metode terlebih dahulu (misalnya content based) , kemudian hasil dari perhitungan tersebut dihitung dengan metode berikutnya (misalnya collaborative filtering) (Hayati, 2011). Penggabungan ini digambarkan pada gambar 2.2 berikut.



**Gambar 2.2** Sequential Combination

#### 2.1.4.3 Penggabungan secara Item-based Clustering Hybrid Method (ICHM)

Penggabungan ini mengintegrasikan informasi item dan rating pengguna untuk menghitung kemiripan item-item (Hayati, 2011). Penggabungan ini digambarkan pada gambar 2.3.



**Gambar 2.3** Item-based Clustering Hybrid Method

#### 2.1.5 Cosine Similarity

Menurut Manning, Raghavan, dan Schutze (2009), cosine similarity digunakan untuk mengukur kedekatan antara dua vektor. Selain menggunakan cosine, untuk mengukur kedekatan antar vektor dengan menggunakan fungsi similaritas. Beberapa fungsi similaritas yang sering dijumpai adalah Jaccard, Overlap, Assymmetric, Minowski Distance, Pearson Corrlation, dan Cosine. Untuk tujuan klastering yang baik adalah Cosine Similarity. Berikut adalah persamaan dari metode Cosine Similarity :



$$\text{Similarity}(x,y) = \frac{\sum_{i=1}^I x_i y_i}{\sqrt{\sum_{i=1}^I x_i^2 \cdot \sum_{i=1}^I y_i^2}}$$

dimana : x dan y adalah vector yang berbeda.  $x_i$  = term i yang ada pada vector x  $y_i$  = term I yang ada pada vector y Pada hasil perhitungan cosine similarity, hasil perhitungan tertinggi yang menjadi vektor yang terdekat dan vektor yang ingin dibandingkan. Pada kasus SIREP, nilai tertinggi adalah rekomendasi pekerjaan yang akan berikan kepada user.

### 2.1.6 Sum Square Error (SSE)

Sum Square Error adalah proses perhitungan yang bertujuan untuk menghitung seberapa besar error kemungkinan yang bisa terjadi pada sistem. Berikut adalah rumus perhitungan SSE :

$$SSE = \sum_{i=1}^n (x_i - \bar{x})^2$$

dimana:

$X_i$  = nilai yang akan dihitung seberapa besar errornya dan  $\bar{X}$  = nilai rata-rata dalam satu cluster dimana  $X_i$  berada Pada hasil perhitungan SSE semakin kecil nilai SSE-nya akan semakin baik. Sebaliknya, semakin besar nilai SSEnya, maka semakin buruk.

### 2.1.7 Dunn Index

Dunn index adalah salah satu pengukuran cluster validity yang diajukan oleh J.C.Dunn. Cluster validity ini berlandaskan pada fakta bahwa klaster yang terpisah itu biasanya memiliki jarak antar klaster yang besar dan diameter intra klaster yang kecil (Gita & Saikhu, n.d.). Hasil perhitungan dunn index, semakin besar semakin baik.

$$D(C) = \frac{\min_{C_k, C_l \in C, C_k \neq C_l} (\min_{i \in C_k, j \in C_l} \text{dist}(i, j))}{\max_{C_m \in C} \text{diam}(C_m)}$$

### 2.1.8 Silhouette

Silhouette merupakan validasi yang digunakan untuk menentukan baik atau buruknya kluster yang dibentuk. Dalam silhouette memiliki rentang -1 sampai 1. Dikatakan kluster itu baik, apabila hasil perhitungan silhouette semakin mendekati 1. Sebaliknya, kluster dikatakan buruk, apabila silhouette mendekati -1 (Gita & Saikhu, n.d.).

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Nilai a adalah rata-rata dari jarak objek i ke objek-objek lain yang berada pada satu kluster. Sedangkan nilai b adalah minimal dari rata-rata 6 jarak objek i ke objek-objek lain yang berada pada kluster yang berbeda.

## 2.2 Kajian Induktif

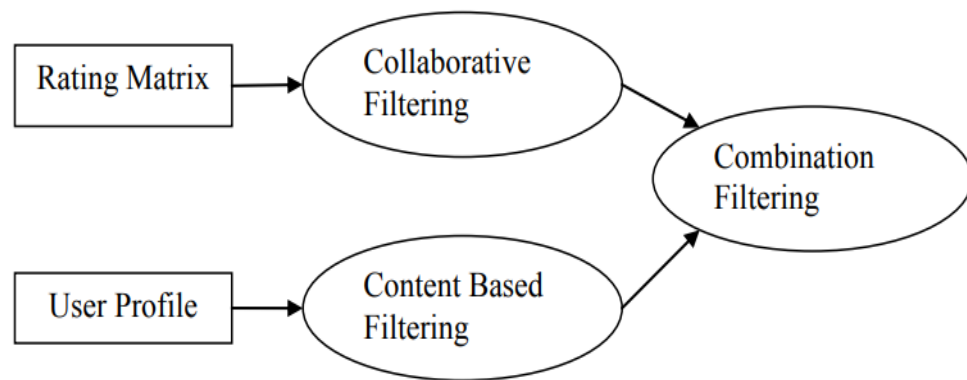
Tinjauan pustaka tersebut adalah hasil penelitian terdahulu tentang informasi hasil penelitian yang telah dilakukan sebelumnya dan menghubungkan dengan masalah yang sedang diteliti.

Menurut Yulisna Gita Hapsari, Agung Toto Wibowo, ZK Abdurahman Baizal (2015) dalam jurnal dengan judul Analisis dan Implementasi Sistem Rekomendasi Menggunakan *Most-Frequent Item* dan *Association Rule Technique* maka dapat ditarik kesimpulan yang menghasilkan sebuah teknik pendekatan paling sukses untuk membangun sistem rekomendasi ini salah satunya adalah dengan menggunakan *collaborative filtering*. Dari hasil penelitian sebelumnya, *association rule* memiliki nilai performansi tertinggi dalam menentukan rekomendasi bila dibandingkan dengan algoritma *matrix factorization* dan algoritma *item-based recommendation*. Sedangkan teknik *most-frequent item* secara sederhana dapat digunakan untuk menghasilkan rekomendasi berdasarkan item yang paling sering digunakan oleh *user* lain (*neighbor*). Pada penelitian ini dibuat sebuah sistem rekomendasi yang membandingkan antara 2 metode yaitu *association rule* dan *most-frequent item*. Sistem rekomendasi ini diterapkan pada data film dan menghasilkan rekomendasi berupa data item yang dapat ditonton oleh user. Hasil dari penelitian yang telah dilakukan adalah bahwa nilai *precision* yang dihasilkan dari metode *association rule* lebih tinggi dari nilai *precision* pada *most frequent item* pada semua skenario pengujian yang dilakukan. Nilai *precision* dari *association rule* adalah 0,38 sedangkan nilai *precision most-frequent item* adalah 0,23.

### BAB III

#### METODE PENELITIAN

Penelitian ini menggunakan data-data teks pada dataset di dalam *movielens*. Pada tahap ini akan dirancang sebuah model *recommender system* yang dibangun dengan menggunakan metode *switching hybrid recommender system*. Metode ini dibangun dengan menggunakan dua pendekatan, yaitu *collaborative filtering* dan *content based filtering* yang akan digunakan untuk menentukan prediksi.



Gambar 3.1 Perancangan *switching hybrid*

Masing-masing teknik sistem rekomendasi memiliki kelebihan dan kelemahannya tersendiri. Karenanya, sistem rekomendasi *hybrid* yang menggabungkan kekuatan dari model dan algoritma yang berbeda untuk mengatasi masalah-masalah yang telah disebutkan di atas menjadi target penelitian yang sedang marak dikembangkan (Jannach et al, 2011).

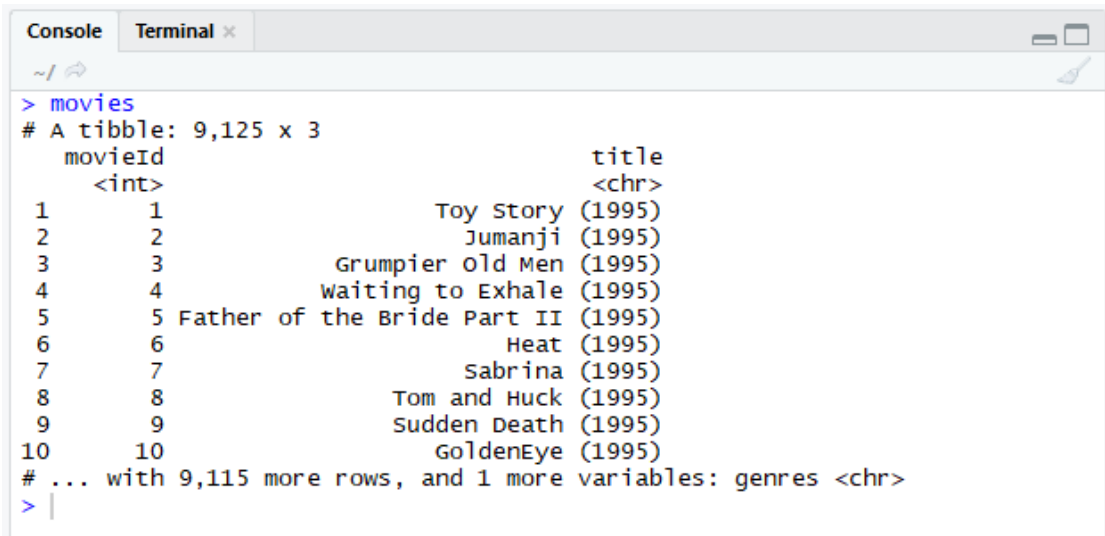
## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Pengumpulan Data

##### 4.1.1. Membaca Data

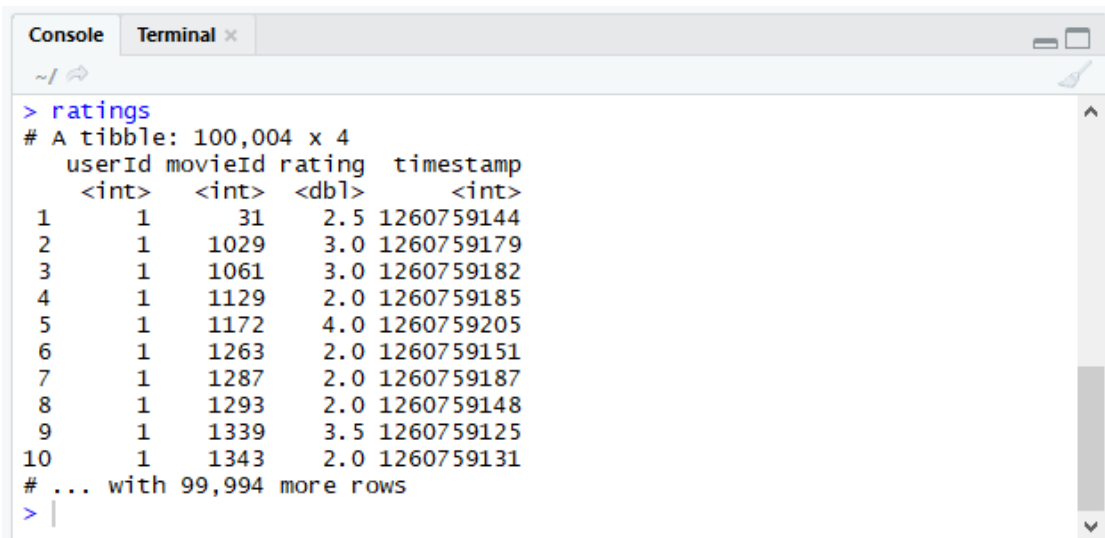
Sebelum melakukan pengolahan data, terlebih dahulu kita baca data yang akan kita olah nantinya dari dataset movielens. Dimulai dari melihat data “movies” seperti yang ditampilkan pada Gambar 4.1.



```
> movies
# A tibble: 9,125 x 3
  movieId title
  <int>   <chr>
1     1 Toy Story (1995)
2     2 Jumanji (1995)
3     3 Grumpier Old Men (1995)
4     4 Waiting to Exhale (1995)
5     5 Father of the Bride Part II (1995)
6     6 Heat (1995)
7     7 Sabrina (1995)
8     8 Tom and Huck (1995)
9     9 Sudden Death (1995)
10    10 GoldenEye (1995)
# ... with 9,115 more rows, and 1 more variables: genres <chr>
>
```

Gambar 4.1. Dataframe Movies

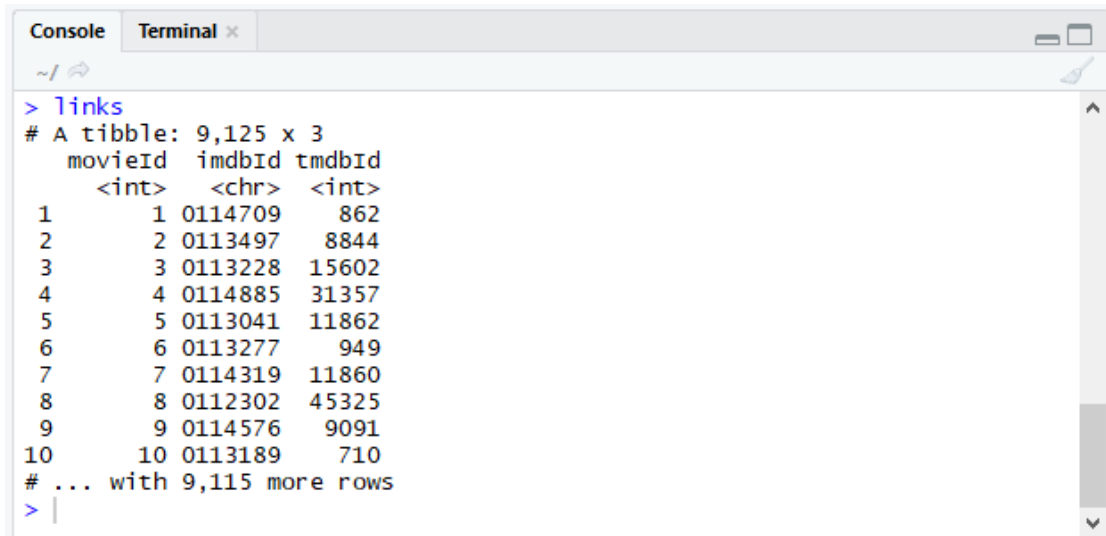
Kemudian melihat data “ratings” seperti yang ditampilkan pada Gambar 4.2.



```
> ratings
# A tibble: 100,004 x 4
  userId movieId rating timestamp
  <int>   <int>   <dbl>   <int>
1     1     31    2.5 1260759144
2     1    1029    3.0 1260759179
3     1    1061    3.0 1260759182
4     1    1129    2.0 1260759185
5     1    1172    4.0 1260759205
6     1    1263    2.0 1260759151
7     1    1287    2.0 1260759187
8     1    1293    2.0 1260759148
9     1    1339    3.5 1260759125
10    1    1343    2.0 1260759131
# ... with 99,994 more rows
>
```

Gambar 4.2. Dataframe Ratings

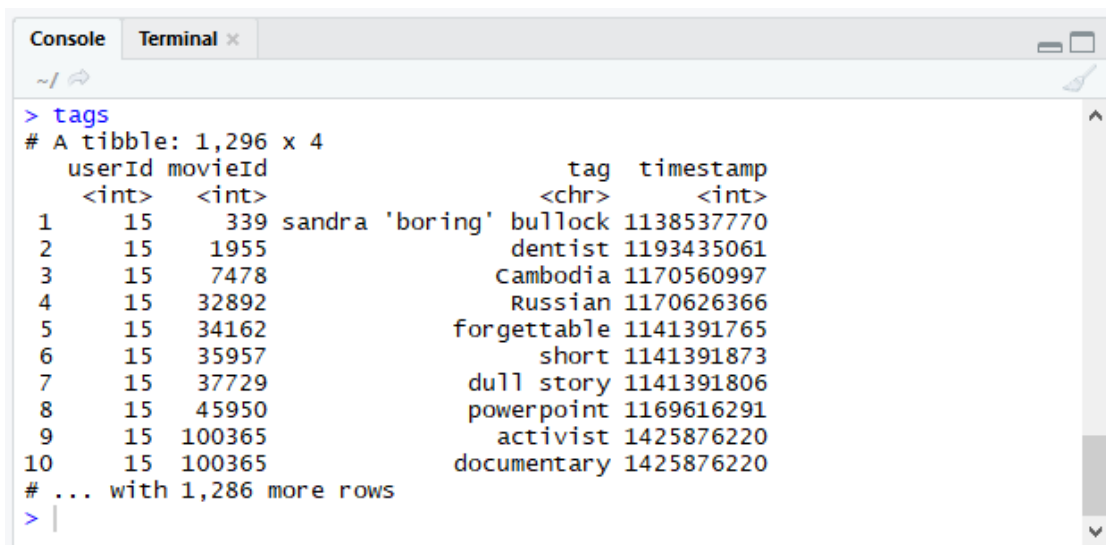
Kemudian melihat data “links” seperti yang ditampilkan pada Gambar 4.3.



```
> links
# A tibble: 9,125 x 3
  movieId  imdbId tmdbId
  <int>    <chr>   <int>
1       1  0114709     862
2       2  0113497    8844
3       3  0113228   15602
4       4  0114885   31357
5       5  0113041   11862
6       6  0113277     949
7       7  0114319   11860
8       8  0112302   45325
9       9  0114576    9091
10      10  0113189     710
# ... with 9,115 more rows
> |
```

Gambar 4.3. Dataframe Links

Kemudian melihat data “tags” seperti yang ditampilkan pada Gambar 4.4.



```
> tags
# A tibble: 1,296 x 4
  userId movieId tag timestamp
  <int>   <int>   <chr>    <int>
1     15     339 sandra 'boring' bullock 1138537770
2     15     1955 dentist 1193435061
3     15     7478 Cambodia 1170560997
4     15    32892 Russian 1170626366
5     15    34162 forgettable 1141391765
6     15    35957 short 1141391873
7     15    37729 dull story 1141391806
8     15    45950 powerpoint 1169616291
9     15   100365 activist 1425876220
10    15   100365 documentary 1425876220
# ... with 1,286 more rows
> |
```

Gambar 4.4. Dataframe Tags

## 4.2 Script dan Hasil Analisa Recommender Movie

```
install.packages("recommenderlab")
```

```
library(datasets)
```

```
library(recommenderlab) # package being evaluated
```

```
library(ggplot2) # For plots
```

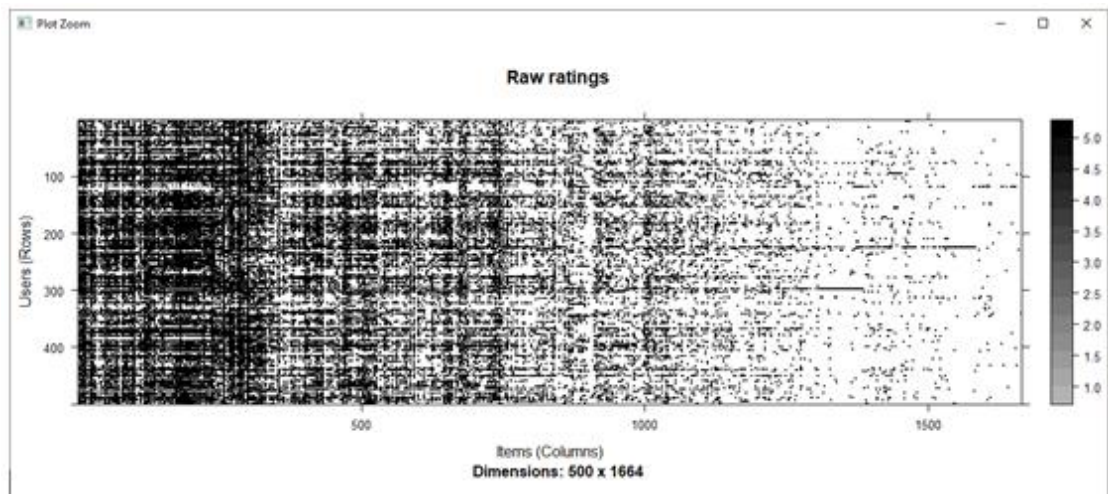
```
# Load the data we are going to work with
```

```
data(MovieLense)
```

```
MovieLense
```

```
Console Terminal x
~/
> data(MovieLense)
> MovieLense
943 x 1664 rating matrix of class 'realRatingMatrix' with 99392 ratings.
> |
```

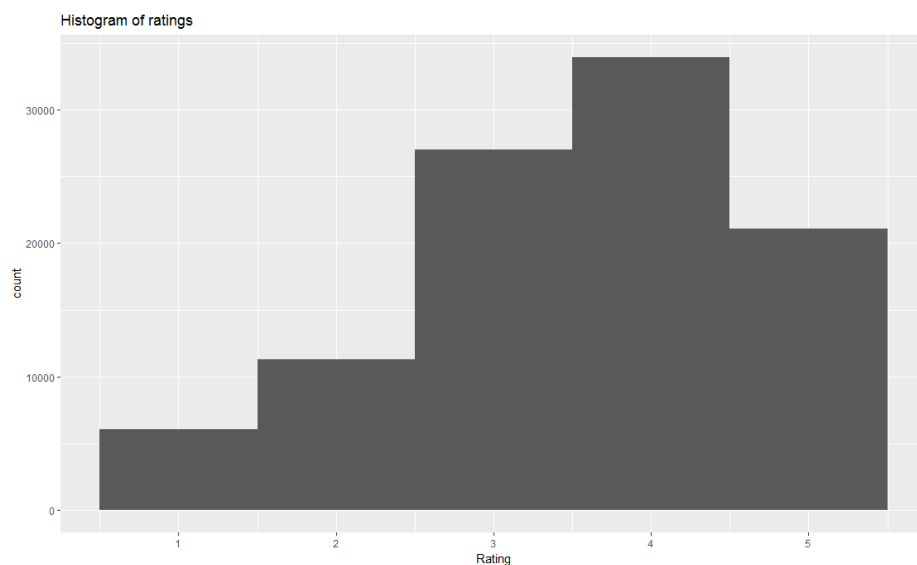
```
image(sample(MovieLense, 500), main = "Raw ratings")
```



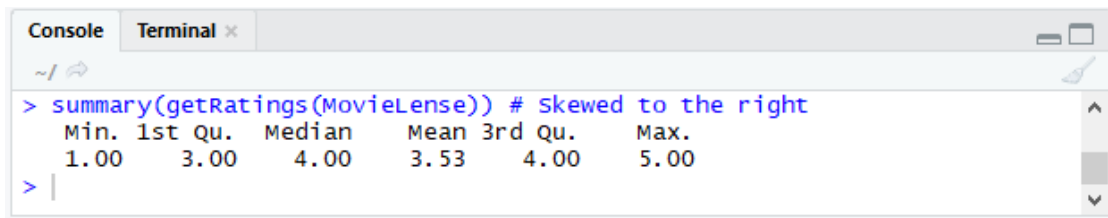
```
# Visualizing ratings
```

```
qplot(getRatings(MovieLense), binwidth = 1,
```

```
main = "Histogram of ratings", xlab = "Rating")
```



```
summary(getRatings(MovieLense)) # Skewed to the right
```



```
Console Terminal x
~/
> summary(getRatings(MovieLense)) # skewed to the right
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.00   3.00   4.00   3.53   4.00   5.00
> |
```

```
# How many movies did people rate on average
```

```
qplot(rowCounts(MovieLense), binwidth = 10,
```

```
main = "Movies Rated on average", xlab = "# of users", ylab = "# of movies rated")
```

```
# Seems people get tired of rating movies at a logarithmic pace. But most rate some.
```

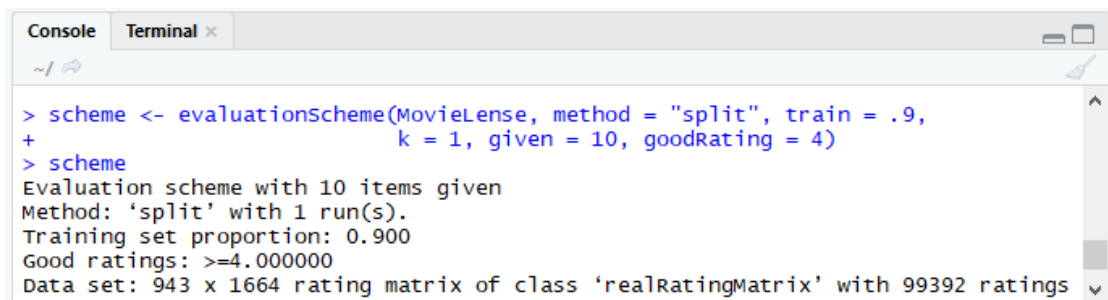
```
recommenderRegistry$get_entries(dataType = "realRatingMatrix")
```

```
# We have a few options
```

```
# Let's check some algorithms against each other
```

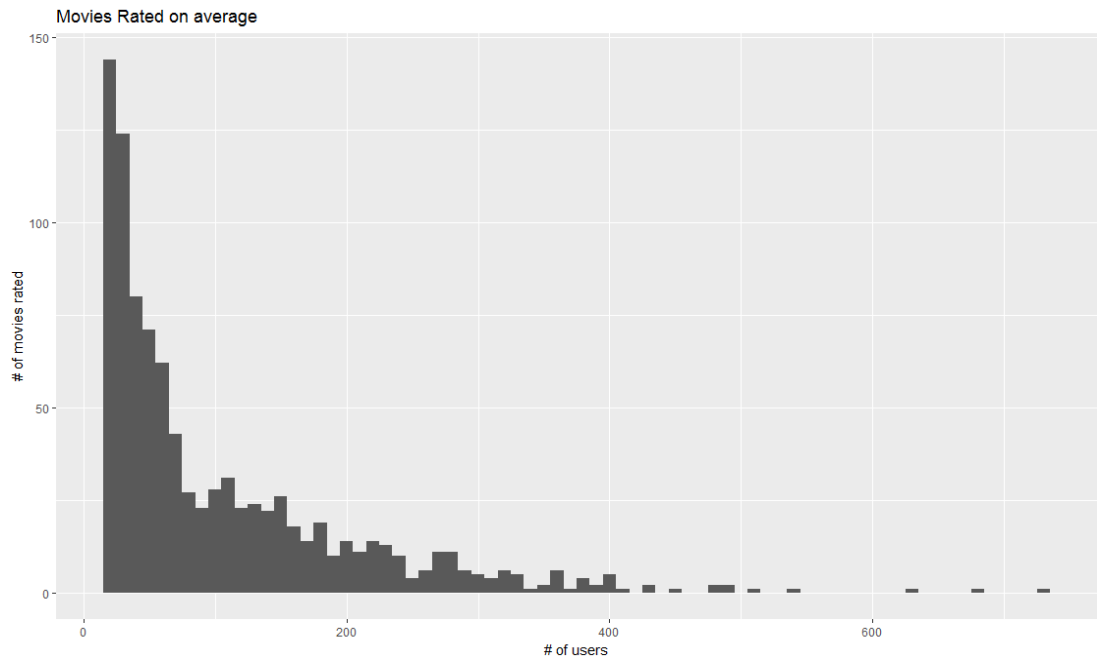
```
scheme <- evaluationScheme(MovieLense, method = "split", train = .9, k = 1, given = 10, goodRating = 4)
```

```
scheme
```



```
Console Terminal x
~/
> scheme <- evaluationScheme(MovieLense, method = "split", train = .9,
+                             k = 1, given = 10, goodRating = 4)
> scheme
Evaluation scheme with 10 items given
Method: 'split' with 1 run(s).
Training set proportion: 0.900
Good ratings: >=4.000000
Data set: 943 x 1664 rating matrix of class 'realRatingMatrix' with 99392 ratings
```

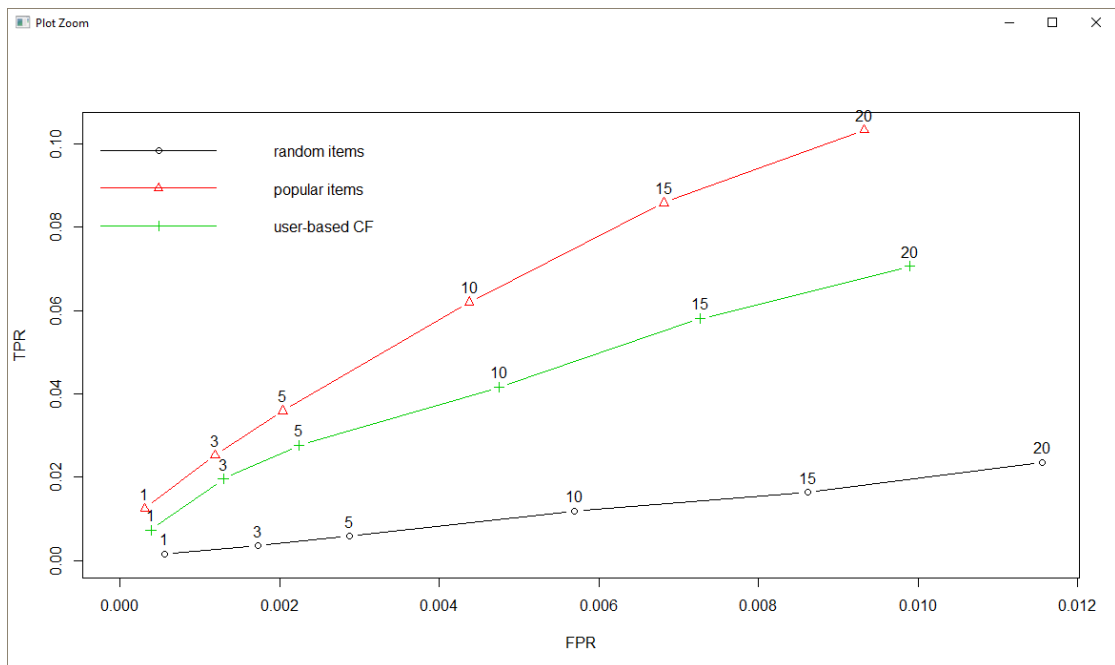
```
algorithms <- list("random items" = list(name="RANDOM", param=list(normalize = "Z-score")), "popular items" = list(name="POPULAR", param=list(normalize = "Z-score")), "user-based CF" = list(name="UBCF", param=list(normalize = "Z-score", method="Cosine", nn=50, minRating=3)))
```



*# run algorithms, predict next n movies*

*results <- evaluate(scheme, algorithms, n=c(1, 3, 5, 10, 15, 20))*

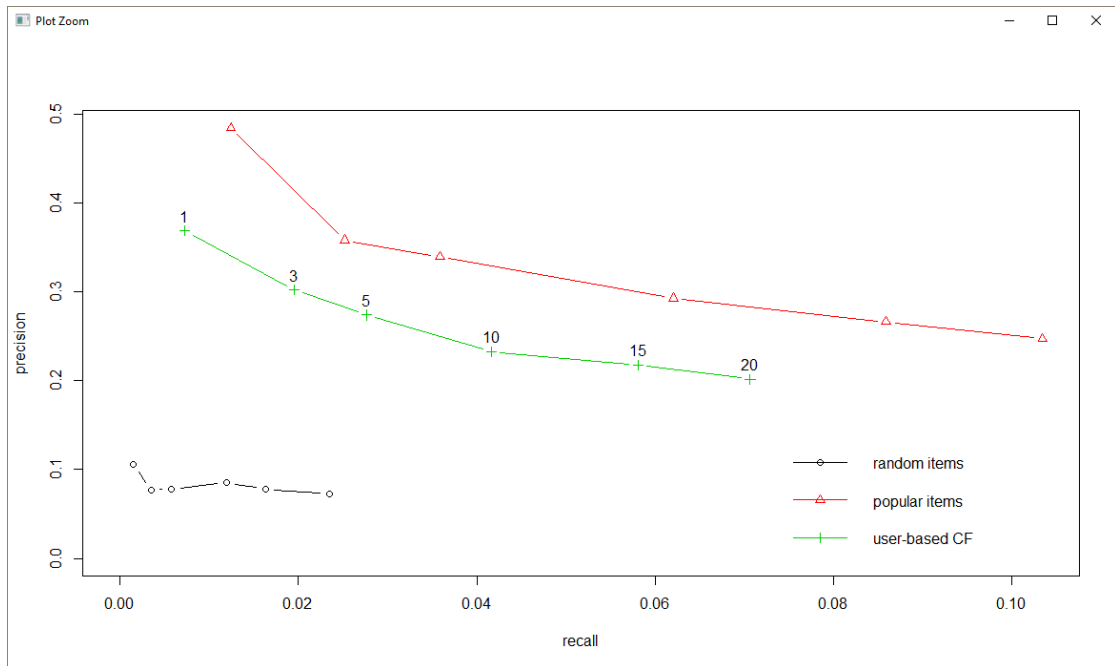
*plot(results, annotate = 1:4, legend="topleft")*





# See precision / recall

`plot(results, "prec/rec", annotate=3)`



## **BAB V**

### **KESIMPULAN**

Sistem rekomendasi (*recommendation system*) merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna. Sistem rekomendasi juga pada umumnya ditujukan untuk individu yang kekurangan pengalaman atau kompetensi yang cukup untuk mengevaluasi banyaknya jumlah alternatif item yang ada pada suatu kasus tertentu. Recommendation system banyak digunakan untuk membuat prediksi seperti buku, music dalam film. Model *recommender system* yang dibangun dengan menggunakan metode *switching hybrid recommender system*. Metode ini dibangun dengan menggunakan dua pendekatan, yaitu *collaborative filtering* dan *content based filtering* yang akan digunakan untuk menentukan prediksi. Hal ini dikarenakan pada metode *hybrid recommender system* item yang belum dirating user lain, akan dihitung prediksi ratingnya menggunakan *content based filtering*. Dengan menggunakan system rekomendasi ini kita dapat melihat data yang belum pernah kita lihat untuk membantu kita dalam memutuskan atau memilih sesuatu yang masih awam bagi kita.

## DAFTAR PUSTAKA

- Yulisna Gita Hapsari, Agung Toto Wibowo, ZK Abdurahman Baizal. 2015. Analisis dan Implementasi Sistem Rekomendasi Menggunakan *Most-Frequent Item* dan *Association Rule Technique*. Universitas Telkom.
- Isnaini Nurul Khasanah, Ade Romadhony, Erda Guslinar Perdana. 2012. *Analisis dan Implementasi Switching Hybrid Recommender System Menggunakan Support Vector Machines Classifier dan Collaborative Filtering Switching Hybrid Recommender System Analysis and Implementation with Support Vector Machines Classifier and Collaborat*. Universitas Telkom.
- RPubs. 2016. *Building a Movie Recommendation System*. Diperoleh 29 Desember 2017, dari <https://rpubs.com/jeknov/movieRec>.
- Kungfumas. 2016, 13 Desember. *Program Recommender Movie*. Diperoleh 29 Desember 2017, dari <https://kungfumas.wordpress.com/page/6/>.