



Rekomendasi Berdasarkan Nilai *Pretest* Mahasiswa Menggunakan Metode *Collaborative Filtering* dan *Bayesian Ranking*

Brilliant Stefani [✉], Teguh Bharata Adji, Sri Suning Kusumawardani dan Indriana Hidayah

Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada Yogyakarta, Indonesia

Info Artikel

Sejarah Artikel:

Diterima: 07 Mei 2018

Disetujui: 12 Juni 2018

Dipublikasikan: 20 Juli 2018

Keywords:

self-regulated learning, metakognitif, metakognitive support, feedback, pretest (prior knowledge), Collaborative Filtering, Bayesian Ranking, Mean Average Precision, runtime complexity.

Abstrak

Kemampuan Self-Regulated Learning (SRL) dapat ditingkatkan dengan meningkatkan kemampuan kognitif dan metakognitif yang dimiliki oleh siswa. Peningkatan kemampuan metakognitif perlu menyertakan adanya metakognitive support dalam pembelajaran menggunakan e-learning. Salah satunya yakni dalam bentuk pendampingan dengan memberikan feedback kepada siswa setelah siswa melakukan aktifitas tertentu. Paper bertujuan untuk mengembangkan suatu pedagogical agent yang mampu memberikan feedback berupa rekomendasi urutan submateri kepada siswa. Pemberian rekomendasi dilakukan dengan mempertimbangkan nilai pretest (prior knowledge) mahasiswa. Perhitungan dilakukan dengan metode Collaborative Filtering dan Bayesian Ranking. Hasil yang didapatkan setelah dilakukan pengujian menggunakan MAP (Mean Average Precision) adalah metode Item-based memiliki nilai MAP tertinggi yakni 1. Waktu perhitungan yang dibutuhkan masing-masing metode dihitung untuk mengetahui runtime complexity dari metode yang digunakan. Hasil yang didapat adalah metode Bayesian Ranking memiliki waktu paling sedikit yakni 0,002 detik, diikuti oleh Item-based 0,006 detik, User Based 0,226 detik, dan waktu terlama adalah Hybrid yakni 0,236 detik

Abstract

© 2018 Universitas Negeri Semarang

[✉] Alamat korespondensi:

E-mail: fani.mti15@mail.ugm.ac.id

ISSN 2252-6811

PENDAHULUAN

Proses belajar mengajar yang masih didominasi oleh guru atau dosen menyebabkan kemampuan *Self-Regulated Learning* (SRL) atau kesadaran belajar secara mandiri pada mahasiswa tidak dapat berkembang secara maksimal. Salah satu cara untuk meningkatkan kemampuan SRL pada mahasiswa adalah dengan meningkatkan kemampuan kognitif dan metakognitif yang dimiliki. Kemampuan metakognitif sangat erat hubungannya dengan penentuan tujuan (*goal*) dan *sub-goal* selama proses pembelajaran. Terdapat tiga hal yang menjadi aspek penting dalam kemampuan metakognitif yakni, mengetahui apa yang akan dipelajari (*what-to-learn*), bagaimana cara mempelajari (*how-to-learn*), dan kapan waktu untuk mempelajari hal tersebut (*when-to-learn*). Pada proses pembelajaran menggunakan *e-learning* yang menerapkan adanya konsep metakognitif perlu menyertakan *metacognitive support* yang mampu meningkatkan kompetensi belajar siswa melalui instruksi-instruksi yang sistematis.

Pemberian instruksi-instruksi yang sistematis dapat meningkatkan kompetensi belajar siswa jika disertai dengan adanya *feedback* dalam proses belajar mengajar menggunakan *e-learning*. Pemberian *feedback* dapat dilakukan dengan dua cara yakni *directed/ correction* dan *guide metacognitive feedback*. Melalui *directed/ correction* mahasiswa akan langsung diberikan *feedback* berupa penilaian benar atau salah terhadap apa yang dikerjakan, sedangkan *guide metacognitive feedback* memungkinkan mahasiswa untuk mempersiapkan apa yang akan dipelajari meskipun peran dari *metacognitive support* telah dihilangkan.

Beberapa penelitian yang membahas tentang penggunaan *metacognitive support* menyatakan bahwa metakognitif support berpengaruh terhadap performa mahasiswa. Bentuk dari *metacognitive support* yang digunakan berupa pemberian instruksi-instruksi sistematis, pemberian pendampingan selama pembelajaran

(*metacognitive scaffolding*), serta pemberian agen yang memiliki peran sama seperti tutor dalam pembelajaran klasikal.

Penelitian menggunakan peran pendampingan dilakukan melalui media tertentu, dimana mahasiswa dan tutor dapat saling berkomunikasi. Peran pendamping dalam penelitian tersebut masih dilakukan oleh guru atau tutor dalam dunia nyata hanya melibatkan media sebagai perantara. Pada penelitian selanjutnya telah dirancang suatu agen yang dapat diintegrasikan dengan *e-learning* interaktif dan mampu menggantikan peran guru atau tutor selama pembelajaran. Pada penelitian tersebut, agen akan memberikan intruksi-instruksi secara sistematis, memberi bantuan jika siswa mengalami kesulitan, serta memberikan strategi yang harus dikerjakan oleh mahasiswa selama proses pembelajaran.

Permasalahan yang dihadapi pada penelitian-penelitian sebelumnya adalah 1) mahasiswa yang diberikan instruksi oleh system merasa terbebani dan tidak bisa bebas memilih materi yang ingin dipelajari, 2) mahasiswa diberikan alternatif berupa materi lain pada topik berbeda yang dapat dipilih mahasiswa untuk dipelajari, 3) tidak adanya *feedback* dari system yang mampu memberi tahu mahasiswa apakah materi yang dipilih sudah benar atau belum. Perlu adanya *feedback* yang mampu membantu mahasiswa dalam membentuk kebiasaan serta kesadaran dalam mempelajari materi.

Tujuan dari paper ini adalah mengembangkan suatu *pedagogical agent* yang bukan hanya mendampingi mahasiswa melainkan juga mampu memberikan *feedback* berupa rekomendasi kepada mahasiswa berupa urutan submateri yang harus dipelajari. Rekomendasi urutan submateri yang dihasilkan nantinya dapat dijadikan rujukan mahasiswa selama memilih submateri yang diinginkan.

Sistem rekomendasi telah banyak digunakan dalam berbagai bidang, seperti *e-commerce*, kesehatan, musik, *game*, film, maupun dunia pendidikan. Sistem rekomendasi berperan penting dalam menyaring informasi yang

disajikan sesuai kebutuhan *user*. Peran *user* dalam proses *filtering* juga sangat menentukan sistem rekomendasi yang dihasilkan akan efektif ataupun tidak. Pada dunia pendidikan, sistem rekomendasi mampu membuat siswa lebih fokus atau tidak buang-buang waktu untuk mencari materi, membantu siswa belajar sesuai gaya belajar masing-masing, dapat membantu guru dalam memonitoring kebutuhan siswa, serta adanya keterlibatan langsung dari siswa dalam proses pembelajaran.

Beberapa penelitian yang telah dilakukan banyak menggunakan metode *Collaborative Filtering* (CF) dalam memberikan rekomendasi. Hal ini dikarenakan metode CF mampu memberikan rekomendasi berdasarkan kemiripan preferensi kebiasaan *user* terhadap suatu *item* tertentu. Penelitian yang pernah dilakukan sebelumnya menggunakan metode CF untuk memfilter *email* yang diterima *user*. *User* akan memberikan reaksi terhadap *email* yang sudah dibaca dan selanjutnya digunakan untuk melakukan *filtering*. Kelemahan pada penelitian ini terletak pada *user experience* yang masih sedikit sehingga data yang dihasilkan belum akurat.

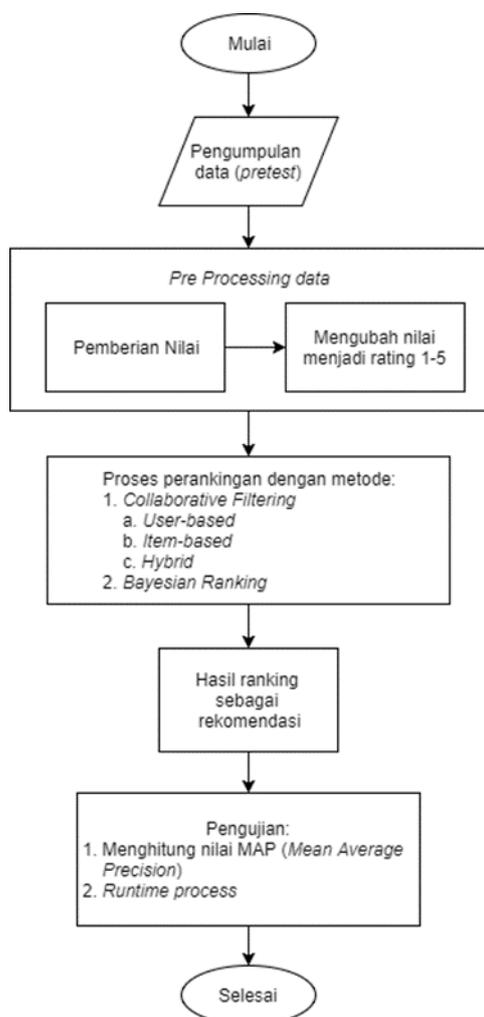
Penelitian lain menggunakan metode CF dalam pemberian rekomendasi untuk menganalisis hubungan antara keterampilan belajar siswa dengan hasil belajar. Peneliti melakukan perbandingan antara penggunaan metode CF tradisional (*user-based* CF) dengan metode *Hybrid* CF yang diusulkan. Hasil yang didapatkan dari penelitian tersebut adalah rekomendasi menggunakan metode yang diusulkan mampu mengurangi kesulitan siswa dalam memilih materi. Terdapat beberapa kelemahan dalam penelitian diantaranya, dibutuhkan waktu yang lama untuk melakukan prediksi materi yang direkomendasikan, belum adanya validasi terhadap keefektifan sistem rekomendasi yang dihasilkan, serta adanya pengaruh perbedaan jenis kelamin yang didominasi oleh perempuan dalam pemberian rekomendasi.

Terdapat beberapa permasalahan yang sering muncul dalam penggunaan metode CF, diantaranya metode CF tidak bisa menangani data dalam jumlah besar karena terkendala pada *runtime complexity*, adanya *cold-start* (data baru yang belum memiliki *rating*), serta adanya kemungkinan *rating* yang diberikan *user* tidak sesuai dengan kondisi sebenarnya. Paper ini akan memberikan rekomendasi urutan submateri kepada mahasiswa dengan mempertimbangkan nilai *pretest* (*prior knowledge*) dari keseluruhan mahasiswa yang ada dalam satu kelas. Proses pemberian rekomendasi nantinya akan dilakukan dengan memanfaatkan metode *Collaborative Filtering* (CF) dan *Bayesian Ranking*.

Paper ini diharapkan mampu menangani permasalahan *cold-start* dengan melibatkan seluruh mahasiswa yang digunakan sebagai objek penelitian menerima *pretest* pada waktu yang sama. Data yang digunakan merupakan data yang sebenarnya dari siswa yakni berupa nilai *pretest* tersebut. Jumlah data yang digunakan sedikit karena hanya dilakukan untuk meneliti satu kelas pembelajaran saja dengan jumlah mahasiswa sebanyak 42 orang.

METODE PENELITIAN

Bahan yang diperlukan adalah data nilai *pretest* mahasiswa pada materi *Searching*. Materi *Searching* yang digunakan meliputi submateri Dasar, *Binary*, *Sequential*, *Binary Tree*, dan *Hash*. Bentuk soal yang digunakan untuk mendapatkan data nilai adalah 15 soal pilihan ganda dan 5 soal uraian singkat. Objek dalam pengambilan data adalah 42 mahasiswa S1 UGM program studi Teknologi Informasi tahun angkatan 2016/ 2017 yang mengambil mata kuliah Algoritma dan Struktur Data.



Gambar 1. Jalannya penelitian

1. Proses Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil data nilai *pretest* pada kelas Algoritma dan Struktur Data dengan materi *Searching*. Pada materi tersebut diambil lima submateri yakni Dasar, *Binary*, *Sequential*, *Binary Tree*, dan *Hash*. Mahasiswa diberikan soal *pretest* berupa 15 soal pilihan ganda dan 5 soal uraian yang mewakili keseluruhan dari submateri yang nantinya akan diajarkan. Masing-masing submateri memiliki jumlah soal yang sama yakni 3 soal untuk pilihan ganda dan 1 soal untuk uraian singkat. Data yang dihasilkan dari *pretest* tersebut adalah data mentah nilai masing-masing butir soal. Tiap soal pilihan ganda diberikan nilai 1 jika benar dan 0 jika salah. Untuk penilaian pada soal

uraian diberikan rentang nilai 0-3 dengan nilai maksimal 3 untuk jawaban yang lengkap dan benar, nilai 0 untuk jawaban yang salah.

2. Preprocessing data

Preprocessing data dilakukan untuk menyesuaikan data yang ada dengan metode yang akan digunakan. Data nilai mahasiswa yang dihasilkan diubah ke alam bentuk *rating* dengan cara mengelompokkan nilai masing-masing sub materi. Data nilai dikonversi ke dalam bentuk *rating* dengan ketentuan rentang nilai 0-20 mendapat *rating* 1, nilai 21-40 mendapat *rating* 2, nilai 41-60 mendapat *rating* 3, nilai 61-80 mendapat *rating* 5, dan nilai 81-100 mendapat *rating* tertinggi yakni 5. Contoh dari masing-masing nilai pada sub materi dapat dijabarkan pada Tabel 1 berikut.

Tabel 1. Nilai pada masing-masing submateri

Mhs	Nilai				
	Dasar	Binary	Sequential	Binary Tree	Hash
1	50.00	50.00	100.00	66.67	100.00
2	50.00	50.00	16.67	33.33	83.33
3	83.33	50.00	50.00	33.33	50.00
.....					
42	50.00	33.33	83.33	16.67	33.33

Perhitungan nilai tiap sub materi dilakukan dengan menjumlahkan seluruh nilai tiap soal yang didapat dibagi dengan total soal selanjutnya dikalikan seratus. Dapat dirumuskan seperti pada Persamaan 1.

$$Nilai = \left(\frac{\text{total nilai tiap soal}}{\text{total soal}} \right) \times 100 \quad (1)$$

Contoh perhitungannya adalah jika mahasiswa memiliki jumlah benar dari keseluruhan soal sebanyak 3 pada submateri Dasar, maka nilai yang diperoleh adalah sebesar 50,00 untuk submateri Dasar dan mendapat *rating* sebesar 3. Penjelasan lebih lanjut data *rating* masing-masing mahasiswa dapat ditunjukkan pada Tabel 2 berikut:

Tabel 2. Data *rating* masing-masing mahasiswa

Mhs	Dasar	Binary	Sequential	Binary Tree	Hash
1	3	3	5	4	5
2	3	3	1	2	5
3	5	3	3	2	3
4	4	3	5	2	2
5	5	4	5	3	2
				
42	3	2	5	1	2

3. Proses Perankingan

Proses perankingan dilakukan dengan memanfaatkan metode *memory-based CF*, yakni *User-based*, *Item-based*, dan *Hybrid* (gabungan *User-based* dan *Item-based*). Metode lain yang digunakan yakni *Bayesian Ranking*. Proses perhitungan pada masing-masing metode menggunakan data *rating* yang telah dihitung sebelumnya.

A. Metode CF

Pada perhitungan dengan metode CF dilakukan dengan tiga cara yakni *User-based*, *Item-based*, dan *Hybrid* (gabungan *User-based* dan *Item-based*). Perhitungan dengan metode CF diawali dengan mencari kemiripan antara masing-masing *user* dan *item* dengan menggunakan perhitungan *Pearson Correlation*. Persamaan *Pearson Correlation* yang digunakan adalah seperti yang ditunjukkan pada Persamaan 2 berikut.

$$P(Ua Ui) = \frac{\sum_{j=1}^m (Ra.j - \bar{Ra})(Ri.j - \bar{Ri})}{\sqrt{\sum_{i=1}^m (Ra.j - \bar{Ra})^2} \sqrt{\sum_{i=1}^m (Ri.j - \bar{Ri})^2}}$$

Keterangan:

Ra j = *rating* yang diberikan oleh *user* Ua kepada *item* j,

Ri j = *rating* yang diberikan oleh *user* Ui kepada *item* j,

\bar{Ra} = rata-rata *rating* dari *user* Ua,

\bar{Ri} = rata-rata *rating* dari *user* Ui,

m = jumlah *user*.

Masing-masing metode akan dijabarkan sebagai berikut:

1) User-based

Perhitungan menggunakan *User-based* didasarkan pada *rating* yang diberikan oleh masing-masing *user*. Proses perhitungan dilakukan dengan menggunakan Persamaan 3 berikut:

$$Pa.i = \bar{Ra} + \frac{\sum_{u=1}^n (Ru.i - \bar{Ra})xP(Ua.Uu)}{\sum_{u=1}^n P(Ua.Uu)}$$

Keterangan:

\bar{Ra} = rata-rata *rating* yang diberikan oleh *user* Ua,

R u.i = *rating* yang diberikan *user* Uu terhadap *item* i,

P(Ua.Uu) = kemiripan antar *user*,

n = jumlah *user*.

2) Item-based

Perhitungan menggunakan *Item-based* dilakukan dengan memperhatikan *rating* pada masing-masing *item*. Proses perhitungan dilakukan dengan Persamaan 4 berikut:

$$Pa.i = \bar{Ra} + \frac{\sum_{u=1}^n (Ru,i - \bar{Ra})xP(Ca.Cu)}{\sum_{u=1}^n P(Ca.CU)}$$

Keterangan:

\bar{Ra} = rata-rata *rating* masing-masing *item* i,

Ru,i = *rating* yang diberikan *user* Uu terhadap *item* i,

P(Ca.Cu) = korelasi per materi,

n = jumlah *user*.

Pada perhitungan *Item-based* besarnya *rating* pada masing-masing *item* menjadi ukuran untuk melakukan perhitungan. Rata-rata dari masing-masing *item* pada 42 mahasiswa yang ada nantinya dimasukkan ke dalam rumus perhitungan untuk masing-masing materi seperti pada perhitungan dengan *Item-based* sebelumnya.

3) Hybrid

Pada perhitungan dengan metode *Hybrid* dilakukan penggabungan antara *User-based* dan *Item-based* seperti ditunjukkan pada Persamaan 5 berikut:

$$Pa, i = \bar{Ra} + \frac{\sum_{u=1}^n (Ru, i - \bar{Ra}) \times P(Ua, Uu) \times P(Ca, Cu)}{\sum_{u=1}^n P(Ca, Cu)}$$

Keterangan:

\bar{Ra} = rata-rata *rating* pada *user* Ua ,

Ru, i = *rating* yang diberikan *user* Uu terhadap *item* i ,

$P(Ua, Uu)$ = kemiripan antar *user*,

$P(Ca, Cu)$ = korelasi per materi,

n = jumlah *user*

Pada perhitungan dengan metode *Hybrid* nilai *similarity* (kemiripan) antar *user* dan korelasi antar *item* menjadi acuan dalam perhitungan.

B. Metode Bayesian ranking

Perhitungan dengan metode *Bayesian Ranking* lebih menekankan pada jumlah *user* pada masing-masing *rating*. Langkah pertama yang dilakukan adalah dengan menghitung jumlah *user* pada masing-masing *rating* yang selanjutnya digunakan untuk menghitung nilai *naïve average* pada masing-masing materi. Perhitungan *naïve average* dapat dicontohkan dalam pemberian *rating* pada suatu produk, misalnya suatu produk memiliki *rating* 5 sebanyak 5, *rating* 4 sebanyak 3, *rating* 3 sebanyak 3, *rating* 2 sebanyak 1, dan *rating* 1 sebanyak 3. Perhitungannya adalah sebagai berikut:

$$A: \frac{5 \times 5 + 4 \times 3 + 3 \times 3 + 2 \times 1 + 1 \times 3}{5 + 3 + 3 + 1 + 3} = \frac{25 + 12 + 9 + 2 + 3}{15} \\ = \frac{51}{15} = 3,4$$

Berdasarkan perhitungan di atas maka didapatkan nilai *naïve average* sebesar 3,4.

Perhitungan selanjutnya yakni mencari nilai *Overall Average* dengan perhitungan yang hampir sama dengan mencari nilai *naïve average* sebelumnya. Jika pada *naïve average* jumlah *user* yang digunakan hanya pada satu produk saja, maka untuk *overall average* jumlah *user* didapatkan dari keseluruhan produk yang akan dihitung rankingnya. Contoh perhitungan pada *overall average* dapat ditunjukkan pada Tabel 3 berikut:

Tabel 3 Contoh *rating* pada produk

Produk	Rating					Total
	5	4	3	2	1	
A	5	3	3	1	3	15
B	3	2	4	2	1	12
C	4	2	1	5	2	14
Total	12	7	8	8	6	41

Perhitungan *overall average* dilakukan dengan cara mengalikan nilai total tiap *rating* dengan besar masing-masing *rating* yang diberikan dan dibagi dengan keseluruhan jumlah pemberi *rating*. Contoh perhitungannya adalah sebagai berikut:

$$\text{Overall Avg} = \frac{12 \times 5 + 7 \times 4 + 8 \times 3 + 8 \times 2 + 6 \times 1}{12 + 7 + 8 + 8 + 6} \\ = \frac{60 + 28 + 24 + 16 + 6}{41} = 3,268$$

Langkah terakhir dari perhitungan ranking dengan *Bayesian* adalah menghitung nilai *Bayesian Ranking* tersebut dengan Persamaan 6 berikut:

$$\frac{\text{OveNum} \times \text{OveAvg} + \text{IndNum} \times \text{IndAvg}}{\text{OveNum} + \text{IndNum}}$$

Keterangan:

OveNum = Overall number,

OveAvg = Overall average,

IndNum = Individual number,

IndAvg = Individual average.

4. Proses Rekomendasi

Proses rekomendasi didasarkan pada hasil dari perhitungan ranking pada masing-masing materi. Submateri yang memiliki nilai paling tinggi dari hasil perhitungan menggunakan metode yang diajukan dianggap sebagai urutan pertama untuk direkomendasikan kepada mahasiswa. Misalnya pada perhitungan dengan metode *Bayesian Ranking* didapatkan nilai akhir 3,303 untuk produk A, 3,107 untuk produk B, dan 3,1605

untuk produk C. Berdasarkan nilai tersebut maka diperoleh urutan rekomendasi yakni produk A sebagai peringkat pertama, produk C peringkat kedua, dan produk B sebagai peringkat terakhir berdasarkan pada *rating* yang diberikan oleh *user* kepada masing-masing produk tersebut.

5. Pengujian

Pengujian dilakukan untuk mengetahui metode mana yang paling sesuai atau akurat digunakan dalam pemberian grup rekomendasi dengan data *pretest* mahasiswa. Pada penelitian yang dilakukan akan digunakan dua jenis pengujian yakni pengujian algoritme dan perhitungan nilai MAP (*Mean Average Precision*).

A. Pengujian algoritme

Pada pengujian algoritme akan dilakukan pengujian terhadap waktu yang diperlukan pada proses perhitungan (*runtime*). Hasil perhitungan tersebut akan menunjukkan algoritme mana yang paling lama atau singkat dalam melakukan perhitungan.

Perhitungan *runtime* algoritme dilakukan dengan menambahkan *library pygraph* pada kode perhitungan masing-masing algoritme. *Library* tersebut berfungsi untuk mengetahui *runtime* proses dari perhitungan dan alur mana saja yang dilalui untuk memperoleh hasil perhitungan. Semakin sedikit waktu yang dibutuhkan dalam melakukan perhitungan maka semakin baik pula metode yang digunakan.

B. Perhitungan nilai MAP

Perhitungan MAP dilakukan untuk mengetahui tingkat presisi dari metode yang digunakan. MAP akan dihitung dengan cara membandingkan hasil rekomendasi dengan susunan atau ranking yang ditentukan oleh *expert judgment* (dosen mata kuliah). Nilai MAP adalah 0-1, semakin mendekati 1 maka hasil rekomendasi semakin presisi dengan susunan ranking yang sebenarnya, hasil MAP mendekati 0 maka bernilai sebaliknya. Persamaan 7 menunjukkan langkah perhitungan dari MAP.

$$MAP = \frac{\sum_q^Q AveP(q)}{Q}$$

Keterangan:

Q = jumlah seluruh *query*,

q = *query* yang dihitung sekarang.

HASIL DAN PEMBAHASAN

A. Rekomendasi

1. Menghitung nilai *similarity* dan korelasi

a. *Similarity*

Hasil perhitungan *similarity* pada:

Tabel 4. *Similarity* antar mahasiswa

Mhs	1	2	3	...	40	41	42
1	1	0,88 2	0,90 4	...	0,82 4	0,92 9	0,89 8
2	0,8 82	1	0,88 7	...	0,67 6	0,77 1	0,70 4
3	0,9 04	0,88 7	1	...	0,90 5	0,90 6	0,89 7
...
40	0,8 24	0,67 6	0,90 5	...	1	0,92 1	0,95 3
41	0,9 29	0,77 1	0,90 6	...	0,92 1	1	0,99 1
42	0,8 98	0,70 4	0,89 7	...	0,95 3	0,99 1	1

a. *Pearson Corellation Coefficient*

Perhitungan korelasi antar *item* digunakan metode PCC dengan hasil seperti pada tabel 5 berikut.

Tabel 5. Korelasi antar *item*

Materi	Dasar	Binary	Seq.	Binary Tree	Hash
Dasar	1	0,251	0,00 7	0,154	-0,034
Binary	0,251	1	0,06 1	0,229	0,162
Sequential	0,007	0,061	0,30 1	0,306	-0,134
Binary Tree	0,154	0,229	- 6	1	0,268
Hash	-0,034	0,162	0,13 4	0,268	1

2. Hasil rekomendasi

Hasil rekomendasi pada masing-masing metode dapat ditunjukkan pada Tabel 6 berikut:

Tabel 6. Perbandingan hasil rekomendasi masing-masing metode

Metode	Urutan Materi (<i>predict value</i>)				
	1	2	3	4	5
<i>User-based</i>	Dasar (168,62)	Sequensial (151,62)	Binary (118,62)	Hash (109,62)	Binary Tree (83,62)
<i>Item-based</i>	Dasar (151,76)	Sequensial (138,236)	Binary (122,01)	Binary Tree (108,97)	Hash (99,58)
<i>Hybrid</i>	Dasar (184,45)	Sequensial (157,591)	Binary (112,87)	Hash (105,11)	Binary Tree (42,30)
<i>Bayesian ranking</i>	Dasar (3,186)	Sequensial (3,119)	Binary (2,99)	Hash (2,95)	Binary Tree (2,85)

B. Pengujian

1. MAP

Perhitungan MAP dilakukan dengan membandingkan hasil rekomendasi dengan urutan materi menurut *expert judgment* (dosen mata kuliah). Urutan submateri menurut *expert judgment* adalah Dasar, *Sequential*, *Binary*, *Binary Tree*, dan *Hash*. Perhitungan MAP pada masing-masing metode adalah seperti pada Tabel 7 berikut.

Tabel 7. Hasil perhitungan MAP

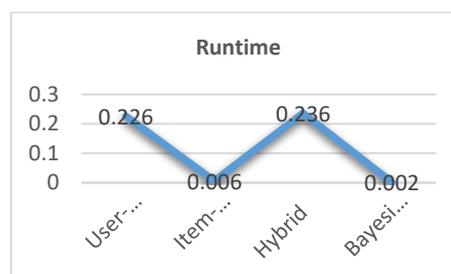
Metode	MAP
<i>User-based</i>	0,87
<i>Item-based</i>	1
<i>Hybrid</i>	0,87
<i>Bayesian Ranking</i>	0,87

Berdasarkan Tabel 7 di atas diketahui bahwa metode *Item-based* memiliki nilai MAP sebesar 1. Hal ini menandakan bahwa metode *Item-based* telah melakukan rekomendasi sesuai dengan urutan yang diberikan oleh *expert judgment* berdasarkan nilai *pretest* yang diperoleh mahasiswa. Sedangkan pada metode lain diketahui bahwa terdapat perbedaan pada urutan submateri yang dihasilkan, yakni pada materi *Hash* dan *Binary Tree*.

2. Runtime

Perbandingan waktu yang dibutuhkan untuk menjalankan perhitungan pada masing-masing metode dapat ditunjukkan pada gambar 2 berikut.

Berdasarkan Gambar 2 dapat dilihat bahwa untuk data sebanyak 42 mahasiswa diperlukan waktu yang relatif singkat dalam melakukan perhitungan. Metode *Bayesian Ranking* memiliki waktu perhitungan paling singkat yakni 0,002 detik untuk melakukan perhitungan rekomendasi urutan sub materi, sedangkan waktu terlama dalam melakukan perhitungan adalah dengan menggunakan metode *Hybrid*.



Gambar 2. Perbandingan *runtime* masing-masing metode

Metode *Bayesian Ranking* melakukan perhitungan dengan menghitung jumlah mahasiswa yang memberikan rating pada masing-masing submateri sehingga waktu perhitungan lebih sedikit. Lamanya waktu perhitungan menggunakan metode *Hybrid* disebabkan oleh lebih banyaknya proses yang dilakukan. Metode *Hybrid* melakukan perhitungan untuk mencari kemiripan antar *user* serta mencari korelasi dari masing-masing sub materi yang ada untuk menghasilkan rekomendasi berdasarkan *user* dan

materi yang dipilih. Pada perhitungan menggunakan metode *Item-based* waktu yang diperlukan juga relatif singkat karena pada metode ini hanya digunakan korelasi pada sub materi yang ada. Jumlah materi yang hanya ada 5 menyebabkan perhitungan menjadi lebih singkat.

SIMPULAN

Kesimpulan dari paper ini adalah metode yang diusulkan dapat digunakan untuk menghitung rekomendasi urutan submateri berdasarkan nilai *pretest* mahasiswa. Berdasarkan hasil perhitungan MAP diketahui bahwa metode *Item-based* memiliki nilai 1 yang artinya hasil rekomendasi urutan submateri sesuai dengan urutan yang sebenarnya menurut *expert judgment*. Sedangkan berdasarkan kecepatan waktu perhitungan, diketahui bahwa metode *Bayesian Ranking* memiliki waktu tercepat jika dibandingkan dengan metode yang lain, diikuti oleh *Item-based*, *User-based*, dan yang terlama adalah *Hybrid*.

Penelitian selanjutnya dapat dilakukan dengan memberikan rekomendasi kepada masing-masing siswa untuk lebih meningkatkan kemampuan metakognitif yang dimiliki. Pemberian rekomendasi dapat dilakukan menggunakan metode yang lain dengan menambahkan jenis data yang digunakan, bukan hanya *pretest* namun dapat pula ditambahkan dengan data kuis-kuis selama pembelajaran. Pemberian *feedback* untuk setiap aktifitas yang dilakukan mahasiswa selama menggunakan aplikasi juga dapat dipertimbangan untuk penelitian selanjutnya, bukan hanya pada saat memilih materi. Guna mengetahui waktu perhitungan yang lebih akurat dapat dilakukan dengan mencoba beberapa jumlah data, dimulai dari data dengan jumlah sedikit hingga banyak.

DAFTAR PUSTAKA

- B. Khosravifar, F. Bouchet, R. Feyzi-Behnagh, R. Azevedo, and J. M. Harley. 2013. "Using intelligent multi-Agent systems to model and foster self-regulated learning: A theoretically-based approach using Markov decision process," *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, pp. 413–420.
- Christopher G. Brinton and M. Chiang. 2017. "The Power of Network: Six Principles That Connect Our Lives." Princeton University Press, United State of America, pp. 113–138.
- Hijriani, A., Muludi, K. & Andini, E.A., 2016. Penyajian Hasil Prediksi Pemakaian Air Bersih Pdam Informasi Geofrafis. *Jurnal Informatika Mulawarman*, 11(2).
- D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. 1992. "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70.
- Kementrian Kesehatan Republik Indonesia, 2014. Pedoman Nasional Pengendalian Tuberkulosis 2014.pdf.
- D. Kravvaris and K. L. Kermanindi. 2015. "Ranking Educational Videos : The Impact of Social Presence," pp. 0–8.
- J. Tan and G. Biswas. 2004. "Feedback for Metacognitive Support in Learning by Teaching Environments,"
- J. won Han, J. choon Jo, H. sung Ji, and H. seok Lim. 2016. "A collaborative recommender system for learning courses considering the relevance of a learner's learning skills," *Cluster Comput.*, vol. 19, no. 4, pp. 2273–2284.
- Pai, F. & Huang, K., 2011. Technological Forecasting & Social Change Applying the Technology Acceptance Model to the introduction of healthcare information systems. *Technological Forecasting & Social Change*, 78(4), pp.650–660. Available at: <http://dx.doi.org/10.1016/j.techfore.2010.11.007>.
- L. Bhatia. 2015. "COPAL – Cognitive Personalized Aid for Learning," *Ccip*, pp. 1–6.
- M. Bannert, M. Hildebrand, and C. Mengelkamp. 2009. "Computers in Human Behavior Effects of a metacognitive support device in learning environments," *Comput. Human Behav.*, vol. 25, no. 4, pp. 829–835.
- M. Chen and H. Li. 2008. "Personalized Recommendation Learning Algorithm for Civil Engineering Curriculum," *2008 Int. Conf. Comput. Sci. Softw. Eng.*, pp. 157–160.
- M. D. Ekstrand. 2011. "Collaborative Filtering Recommender Systems," *Found. Trends® Human-Computer Interact.*, vol. 4, no. 2, pp. 81–173.

- M. Manso-vázquez, M. Llamas-nistal, and S. Member. 2015 "Proposal of a Learning Organization Tool With Support for Metacognition," vol. 10, no. 2, pp. 35–42.
- M. Pratama, J. Lu, S. Anavatti, E. Lughofer, and C. Lim. 2016. "Neurocomputing An incremental meta-cognitive-based scaffolding fuzzy neural network," *Neurocomputing*, vol. 171, pp. 89–105.
- M. Taib, M. Adi, and H. Allali. 2014. "Pedagogical Agent for metacognitive scaffolding in interactive learning environments,".
- M. Taub, R. Azevedo, N. Mudrick, E. Clodfelter, and N. Carolina. 2014. "Can Scaffolds from Pedagogical Agents Influence Effective Completion of Sub-Goals during Learning with a Multi-Agent Hypermedia-Learning Environment?," *Learn. Becom. Pract. Int. Conf. Learn. Sci. 2014, Jun 2014, Boulder, CO, United States. Int. Soc. Learn. Sci. 2*, pp.1052–1056, vol. 2, pp. 1052–1056.
- N. F. Jumaat and Z. Tasir. 2015. "Metacognitive Scaffolding to Support Students in Learning Authoring System Subject," *2015 Int. Conf. Learn. Teach. Comput. Eng.*, pp. 87–90.
- N. N. Liu and Q. Yang. 2008. "EigenRank: A Ranking-oriented Approach to Collaborative Filtering," *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 83–90.
- P. Koskela. 2017. "Comparing Ranking-Based Collaborative Filtering Algorithms To A Rating-based Alternative In Recommender Systems Context," University of Jyväskylä.
- R. Azevedo, A. Witherspoon, A. Chauncey, C. Burkett, and A. Fike. 2009. "MetaTutor: A MetaCognitive Tool for Enhancing Self-Regulated Learning," *Annu. Meet. Am. Assoc. Artif. Intell. Symp. Metacognitive Cogn. Educ. Syst.*, pp. 14–19.
- R. C. Weng and C. Lin. 2011. "A Bayesian Approximation Method for Online Ranking," vol. 12, pp. 267–300.
- V. Aleven, B. McLaren, I. Roll, and K. Koedinger. 2006. "Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor," *Int. J. Artif. Intell. Educ.*, vol. 16, no. 2, pp. 101–128.
- Y. Lu, Z. Qiao, P. Zhang, and L. Guo. 2017. "Ranking-based music recommendation in online music radios," *Proc. - 2016 IEEE 1st Int. Conf. Data Sci. Cyberspace, DSC 2016*, no. c, pp. 614–619