

RANCANGAN SISTEM REKOMENDASI GAME DENGAN MODEL-BASED COLLABORATION FILTERING

Herastia Maharani ^{#1}

[#] Departemen Teknik Informatika Institut Teknologi Harapan Bangsa

Jln. Dipatiukur No. 80-84, Bandung 40132 - Indonesia

¹ herastia@ithb.ac.id

Abstract— Pengguna internet saat ini makin kewalahan menghadapi arus informasi yang makin deras di internet. Begitu banyak produk dan informasi yang disajikan, namun tidak ada panduan mana produk atau informasi yang memang benar-benar dibutuhkan setiap user dan mana yang sebenarnya tidak perlu. Di tengah kebingungan ini, kebanyakan user akhirnya memilih untuk meminta rekomendasi dari lingkungan sekitarnya sebelum memilih atau membeli sebuah produk tertentu. Beberapa situs saat ini mulai menyediakan fitur rekomendasi yang mampu memberikan saran produk mana yang dianggap paling cocok untuk masing-masing user. Salah satu metode yang dapat dipakai untuk merancang sistem rekomendasi ini adalah metode model-based collaborative filtering. Makalah ini berisi rancangan sebuah sistem rekomendasi berdasarkan metode model-based collaborative filtering untuk kasus rekomendasi game. Kebanyakan situs game yang ada hanya menyediakan rekomendasi berdasarkan kesamaan genre dari game yang dimiliki. Dalam rancangan yang disajikan di makalah ini, rekomendasi yang diberikan didasarkan kepada kemiripan antar game dan sejarah transaksi yang pernah dilakukan oleh user.

Keywords— Sistem rekomendasi, game, model-based collaborative filtering

Abstract— Nowadays, internet users are becoming more and more overwhelmed with the phenomenon of information overload. There are so many products and informations available on the internet, but the users have absolutely no guidance on telling which information is important for them and which is not. In this kind of situation, users usually seek for recommendations from their surroundings before they actually choose or purchase a product. Some websites today have already included a recommender system that able to suggest which products are most suitable for certain user. One of the method that commonly used in building a recommender system is model-base collaborative filtering. This paper introduces a design for a game recommender system based on model-based collaborative filtering. Most game websites availbale today only give recommendations based on the category of the games. In the design presented in this paper, the recommendations are based on the similarity between games and the history of user transactions.

Keywords— Recommender system, game, model-based collaborative filtering

I. PENDAHULUAN

Perkembangan internet yang sangat pesat telah membawa banyak perubahan dalam perilaku bisnis dan ekonomi masyarakat. Banyak pihak memandang teknologi internet sebagai peluang bisnis yang sangat besar untuk melakukan perdagangan secara elektronik (*e-commerce*). Produk yang ditawarkan pun beraneka ragam, mulai dari buku, film, album musik, MP3, video, dsb. Saat ini, banyak situs-situs *e-commerce* yang sudah dilengkapi kemampuan untuk memberikan rekomendasi sesuai dengan preferensi pelanggan, dengan tujuan supaya pelanggan tertarik untuk membeli produk yang direkomendasikan, contohnya Amazon.com, CDNow, Reel.com, dsb.

Salah satu jenis situs yang saat ini semakin marak dan banyak peminatnya adalah situs-situs penyedia *game*. Situs-situs ini biasanya menyediakan berbagai jenis *game* untuk komputer maupun *video game*. Untuk beberapa jenis *game* tertentu, umumnya disediakan fasilitas bagi pengunjung untuk memainkan *game* tersebut secara online dengan fitur yang terbatas. Untuk memainkan *game* dengan fitur yang lengkap, pengunjung harus membeli *game* yang diinginkan terlebih dahulu. Beberapa situs memungkinkan pengunjung untuk men-download versi *trial* dari *game* yang diinginkan. Versi *trial* ini biasanya hanya dapat dimainkan untuk jangka waktu tertentu, misalnya 60 menit permainan, 10 kali permainan, dsb. Jika masa percobaannya sudah habis, maka user tidak dapat memainkan *game* itu lagi. Situs *game* seperti ini cukup digemari oleh pengguna internet, walaupun untuk di Indonesia kebanyakan user tidak melakukan pembelian melalui situs internet. User biasanya hanya menggunakan situs-situs ini untuk mencari informasi *game* yang menarik bagi mereka, kemudian mereka membelinya langsung di toko-toko penjual *game*. Contoh situs-situs *game* ini antara lain Yahoo!Games, King.com, Boardgamegeek.com, Game House, dsb.

Walaupun situs-situs *game* ini cukup populer, pengguna situs-situs ini umumnya menemui satu masalah yang sama. Informasi dan jenis *game* yang disediakan oleh situs-situs ini terlalu banyak sehingga justru membuat user menjadi bingung, tidak tahu harus memilih *game* yang mana. User biasanya justru akan bertanya pada teman atau bergabung dalam forum-forum diskusi untuk meminta rekomendasi *game* apa yang

kira-kira cocok baginya. Baru setelah itu dia akan mengunjungi situs *game* di internet untuk mencari atau mencoba *game* tersebut. Tentunya akan jauh lebih mudah bagi user jika situs *game* yang dikunjungi dapat langsung merekomendasikan beberapa *game* yang dianggap cocok bagi dirinya [10, 11]. Selain memudahkan pencarian user, mungkin saja sistem dapat merekomendasikan *game* baru yang sama sekali belum dikenal oleh user dan ternyata cocok dengan selera user tersebut. Fokus utama dari *paper* ini adalah merancang sebuah sistem rekomendasi untuk membantu user memilih *game* yang ditawarkan di situs-situs *game* di internet.

II. SISTEM TERKAIT

Dari sekian banyak situs *game* yang ada di internet, sebagian besar masih memberikan rekomendasi yang sangat sederhana. Rekomendasi yang diberikan umumnya berdasarkan pada *rating* yang diberikan oleh user dan statistik pembelian. *Game* yang direkomendasikan adalah yang memiliki *rating* paling bagus dan/atau yang paling banyak dibeli atau *download* oleh user. Pada beberapa situs, rekomendasi yang diberikan juga dibantu dengan kategorisasi berdasarkan *genre*. Seluruh *game* yang ada di situs tersebut dibagi menjadi beberapa *genre*, seperti *board game*, *word game*, *card game*, dsb. Jika user sedang membuka halaman sebuah *game* yang termasuk *word game*, misalnya Bookworm, sistem akan merekomendasikan *game* yang populer (berdasarkan *rating*) di *genre* yang sama, misalnya Scrabble.

Boardgamegeek.com sudah mengembangkan algoritma rekomendasinya sendiri yang didasarkan pada *rating* dan catatan koleksi *game* yang dimiliki user [9]. Misalnya GameX dimiliki oleh 1% user dan GameY dimiliki oleh 300 user. Kita dapat mengestimasi bahwa sekitar 3 user yang memiliki GameY juga memiliki GameX. Jika ada jauh lebih banyak user pemilik GameY yang juga memiliki GameX, hal ini mengindikasikan bahwa sebagian besar orang yang membeli GameY juga akan menyukai GameX. Kesimpulan ini akan sangat tergantung pada berapa banyak pemilik GameY yang juga memiliki GameX. Jika hanya 4 orang, maka hal ini tentunya bukan merupakan anomali yang mengindikasikan apapun. Tetapi jika ada 60 orang, maka ini merupakan anomali yang patut diperhatikan. Hasil perhitungan berdasarkan koleksi user ini digabungkan dengan *rating* yang diberikan user dan menghasilkan nilai tertentu. Di halaman awal untuk setiap *game*, diberikan suatu daftar berisi *game* yang direkomendasikan. Game-game yang memiliki nilai tertinggi akan ditampilkan dalam daftar rekomendasi. Algoritma ini masih memiliki beberapa kekurangan, diantaranya tidak dapat memberikan rekomendasi yang baik untuk *game* baru atau *game* yang tidak populer (pemiliknya sangat sedikit), selain itu ada tendensi beberapa *game* yang menjadi favorit dan terus-menerus direkomendasikan. Algoritma ini baru diperkenalkan di bulan Januari 2008, sebelumnya rekomendasi yang diberikan hanya berdasarkan masukan dari user. User yang sedang berada di halaman satu *game* tertentu, misal GameX, dapat masuk ke menu rekomendasi dan menambahkan *game* apa saja yang menurut mereka akan disukai oleh orang-orang yang menyukai GameX.

Semakin banyak suatu *game* direkomendasikan, semakin tinggi *rating* dari *game* tersebut. User juga dapat melakukan koreksi jika dia tidak setuju dengan rekomendasi yang diberikan oleh sistem ataupun oleh user lain.

Dalam makalah ini akan digunakan metode yang berbeda dengan yang digunakan oleh situs-situs di atas, yaitu dengan menggunakan *model-based collaborative filtering*.

III. DESKRIPSI UMUM SISTEM

Sistem yang dirancang akan menggunakan metode *model-based collaborative filtering* yang menggunakan kemiripan *item-to-item* untuk menghitung hubungan antar *item* yang berbeda. Alasan utama dipilihnya algoritma ini adalah fakta bahwa pengguna *game* biasanya cenderung memilih *game* yang serupa dengan *game* yang sudah pernah dia beli atau mainkan sebelumnya. Umumnya pengguna *game* sudah memiliki preferensi jenis-jenis *game* tertentu, misalnya *game* strategi, *puzzle*, simulasi, otomotif, dsb. Rekomendasi yang dicari biasanya adalah untuk *game* baru yang belum diketahui user, tetapi tetap dalam lingkup jenis *game* yang memang biasa dia mainkan. Karena itu, algoritma ini akan menganalisis sejarah transaksi user (baik pembelian maupun *download*) untuk mengidentifikasi himpunan *game* yang serupa (*similar item*) dan menggunakan kemiripan ini untuk menghasilkan rekomendasi. Karena jumlah rekomendasi yang diberikan kepada user adalah sebanyak N , algoritma ini disebut juga *model-based top-N recommendation algorithm*.

Algoritma yang digunakan terdiri dari 2 tahap utama. Yang pertama adalah membangun model yang menggambarkan hubungan antara pengguna dan *item*. Sedangkan yang kedua adalah menggunakan model yang dihasilkan di tahap pertama untuk menghasilkan sejumlah rekomendasi bagi user. Secara singkat, cara kerja sistem juga terdiri dari 2 tahapan tersebut. Pertama-tama harus dibangun model keterhubungan antara user dan *game* yang ada berdasarkan sejarah transaksi yang pernah dilakukan oleh user dari situs tersebut. Dari model ini, sistem dapat mengidentifikasi kemiripan antar *game*. Selanjutnya, setiap kali ada user yang masuk ke dalam situs, sistem akan melihat sejarah transaksi user tersebut dan merekomendasikan sejumlah *game* lain yang mirip dengan *game* yang pernah dibeli oleh *download* oleh user tersebut dan belum ada dalam sejarah user tersebut. Rancangan sistem secara rinci akan diuraikan di bab 4.

IV. RANCANGAN SISTEM

Dalam bab ini akan diuraikan rancangan sistem secara rinci, mulai dari fitur/layanan secara umum, input yang diperlukan dari user, proses pembentukan rekomendasi (algoritma yang digunakan), hingga output yang diberikan ke user.

A. Informasi Umum yang Disediakan Bagi User

Karena sistem rekomendasi ini akan diimplementasikan ke situs *game*, berarti sistem yang dibangun akan berbasis web. Walaupun sistem ini menghasilkan rekomendasi berdasarkan kemiripan antar *game*, namun sistem ini tetap menyediakan mekanisme bagi user untuk memberikan *rating* terhadap *game* yang pernah dicoba. *Rating* ini akan digunakan untuk

memberikan informasi *game* apa saja yang *rating*-nya paling bagus secara keseluruhan. Selain itu, untuk setiap *game* yang direkomendasikan ke *member*, akan ditampilkan pula *rating*-nya. Jadi walaupun rekomendasi dibuat berdasarkan kemiripan antar *game* dan bukan berdasarkan *rating* dari user, sistem tetap menampilkan *rating* dari setiap *game* yang direkomendasikan ke seorang *member* sebagai bahan pertimbangan lebih lanjut.

User harus mendaftar terlebih dahulu untuk dapat mencoba *game* secara *online*, *men-download*, atau membeli *game* dari situs ini. Jadi diperlukan input data user untuk registrasi sebagai *member*. Untuk pengunjung yang belum terdaftar sebagai *member*, mereka tetap bisa melihat informasi mengenai *game* yang tersedia di situs ini, meliputi nama *game*, *genre* (strategi, otomotif, dsb), pembuat *game*, tahun rilis, *rating*, ilustrasi *game* dan deskripsi singkat mengenai *game* tersebut. Pengunjung umum juga dapat melihat informasi *game* yang *rating*-nya paling bagus dan *game* yang paling banyak dibeli atau *di-download* (dari statistik transaksi *member*). Untuk pencarian *game*, akan disediakan layanan pencarian berdasarkan *genre* tertentu, judul *game* atau berdasarkan *keyword* tertentu. Semua layanan untuk pengunjung umum tentunya juga akan diberikan pada *member*.

Bagi user yang baru mendaftar (belum pernah melakukan pembelian), ketika dia melakukan login dan masuk ke halaman utama, dia belum mendapat rekomendasi apapun selain informasi *game* dengan *rating* tertinggi dan yang paling banyak dibeli oleh *member* lain. Namun setelah dia memilih satu *game* tertentu dan masuk ke halaman untuk *game* itu, dia akan langsung mendapat rekomendasi berupa daftar *game* yang mirip dengan *game* yang sedang dia akses.

Bagi user yang sudah pernah melakukan transaksi, begitu melakukan *login* dan masuk ke halaman utama, dia akan langsung mendapatkan rekomendasi berdasarkan sejarah *game* yang pernah dia beli atau *download*. Selain itu, sistem akan menyediakan menu untuk menampilkan daftar *game* yang pernah dibeli atau *di-download* oleh user tersebut. Layanan lainnya sama dengan kasus untuk user yang baru mendaftar.

B. Input dari User

Input yang diperoleh dari user terdiri dari :

1. Data pribadi user, diberikan pada saat registrasi
2. *Rating* dari user (opsional)

Member dapat memberikan *rating* untuk *game* yang tersedia dalam sistem. Sistem *rating* yang digunakan menggunakan skala 1-5 (Sangat jelek, Jelek, Biasa, Bagus, Sangat Bagus), namun pecahan 0,5 juga dapat digunakan sehingga nilai seperti 3,5 atau 2,5 juga dimungkinkan. Seperti telah diuraikan di sub bab 4.1, *rating* ini akan ditampilkan sebagai pelengkap hasil rekomendasi dan akan digunakan untuk menampilkan informasi *game* apa saja yang *rating*-nya paling bagus. Yang bisa memberi *rating* hanya user yang sudah terdaftar sebagai *member* saja. Layanan untuk memberi *rating* ini dapat diakses dari halaman utama, atau diisi di halaman awal dari setiap *game*.

3. *Feedback* dari user (opsional)

Rekomendasi yang dihasilkan oleh sistem belum tentu cocok dengan keinginan user. Karena itu disediakan layanan bagi user untuk memberikan *feedback* terhadap hasil rekomendasi untuk menunjukkan apakah dia menyukai rekomendasi yang diberikan atau tidak. Di setiap *game* yang direkomendasikan akan disediakan *checkbox* untuk memilih apakah rekomendasi itu ingin dihilangkan secara permanen atau dibiarkan saja. Jika user tidak menyukai rekomendasi yang diberikan, dia dapat memilih untuk menghilangkan *game* itu dari daftar rekomendasi. *Feedback* ini akan digunakan untuk merevisi hasil rekomendasi sistem di masa mendatang.

C. Algoritma Rekomendasi

Dalam sub bab ini akan dijelaskan algoritma rekomendasi *model-based top-N* yang digunakan. Dalam penjelasan sub bab ini, akan digunakan simbol-simbol sebagai berikut :

- n = jumlah user
- m = jumlah *item* (*game*) dalam sistem
- N = jumlah rekomendasi yang diberikan ke user

Himpunan transaksi yang pernah dilakukan oleh user akan dicatat dalam matriks biner R berukuran $n \times m$, yang disebut matriks *user-item*. R_{ij} akan bernilai satu (1) jika user ke- i pernah membeli *item* ke- j , atau nol (0) jika sebaliknya. User yang akan diberi rekomendasi disebut sebagai user aktif, dan himpunan *item* yang pernah dibeli oleh user disebut himpunan U dan disebut *basket* user.

Permasalahan *top-N recommendation* secara formal didefinisikan sebagai berikut [3] :

Diberikan sebuah matriks *user-item* R dan satu himpunan *item* U yang sudah pernah dibeli oleh user, identifikasikan suatu himpunan *item* terurut X sedemikian sehingga $|X| \leq N$ dan $X \cap U = \emptyset$

Sebagaimana disebutkan di bab 4.3, algoritma ini terdiri dari 2 tahap, yaitu :

1. Pembangunan Model

Algoritma umum untuk membangun model yang diperlukan diberikan di Algoritma 4.1 [3]. Input untuk algoritma ini adalah matriks *user-item* R berukuran $n \times m$ dan sebuah parameter k yang mendefinisikan jumlah kemiripan *item-to-item* yang akan disimpan untuk setiap item. Output dari algoritma ini adalah model itu sendiri yang direpresentasikan dalam bentuk matriks M berukuran $m \times m$, sedemikian sehingga kolom ke- j menyimpan k buah item yang paling mirip dengan item j . Secara spesifik, jika $M_{i,j} > 0$, berarti item ke- i termasuk dalam k buah item yang paling mirip dengan item j dan nilai dari $M_{i,j}$ mengindikasikan derajat kemiripan antara i dan j .

Penggunaan parameter k didasarkan pada pertimbangan performansi dan nilainya akan merepresentasikan *trade-off* antara performansi dan kualitas. Dengan menggunakan nilai k yang kecil, maka matriks M dapat disimpan di memori utama meskipun nilai m sangat besar, Namun demikian, jika nilai k terlalu kecil, maka model yang dihasilkan akan memiliki

informasi yang sangat terbatas untuk menghasilkan rekomendasi, sehingga kualitas rekomendasi akan menurun. Berdasarkan percobaan yang dilakukan di [3], nilai k yang kecil ($10 \leq k \leq 30$) sudah dapat menghasilkan rekomendasi yang bagus dan nilai k yang lebih besar tidak memberikan peningkatan kualitas yang signifikan.

```

for j ← 1 to m do
  for i ← 1 to m do
    if i ≠ j
      then  $M_{i,j} \rightarrow \text{sim}(R_{*,j}, R_{*,i})$ 
      else  $M_{i,j} \rightarrow 0$ 
  for i ← 1 to m do
    if  $M_{i,j} \neq$  among the  $k$  largest values in
       $M_{*,j}$ 
      then  $M_{i,j} \rightarrow 0$ 
return (M)

```

Algoritma 4.1

Algoritma yang digunakan untuk mengkonstruksi matriks M sangat sederhana. Untuk setiap *item* j , algoritma ini akan menghitung kemiripan antara j dan semua *item* lainnya dan menyimpan hasil perhitungan tersebut dalam kolom ke- j dari matriks M . Setelah semua kemiripan antar *item* telah dihitung, langkah berikutnya adalah memeriksa semua entri di kolom ke- j matriks M . Semua entri/element yang tidak termasuk dalam k buah nilai kemiripan terbesar akan diset dengan nilai nol (0). Model akhir yang dihasilkan adalah matriks M yang mengandung maksimal k buah entri bukan nol (*nonzero*) di setiap kolomnya. Algoritma yang diberikan sudah menjamin bahwa sistem tidak akan merekomendasikan *item* yang sudah ada dalam *basket* milik *user* yang sedang aktif.

Untuk menghitung kemiripan antara dua buah *item*, digunakan persamaan *cosine-based similarity*. Jika R adalah matriks *user-item* berukuran $n \times m$, maka kemiripan antara dua *item* i dan j didefinisikan sebagai cosinus dari vektor n -dimensi yang merepresentasikan kolom ke- i dan kolom ke- j dari matriks R , sebagai berikut [3]:

$$\text{sim}(i, j) = \cos(\bar{R}_{*,i}, \bar{R}_{*,j}) = \frac{\bar{R}_{*,i} \cdot \bar{R}_{*,j}}{\|\bar{R}_{*,i}\| \|\bar{R}_{*,j}\|},$$

2. Penerapan Model untuk Menghasilkan Rekomendasi

Algoritma dasar untuk menerapkan model *item-based* diberikan pada Algoritma 4.2 [3]. Input dari algoritma ini adalah model M , vektor U berukuran $m \times 1$ yang menyimpan daftar *item* yang sudah pernah dibeli oleh user yang sedang aktif, dan jumlah rekomendasi yang diinginkan (N). Informasi mengenai user aktif dalam vektor U diisialisasi dengan mengisi nilai $U_i = 1$ jika user telah membeli *item* ke- i , atau $U_i = 0$ jika sebaliknya. Karena dalam sistem yang dirancang user dapat memilih untuk menghilangkan *game* tertentu yang tidak dia sukai dari daftar rekomendasi, maka domain nilai untuk vektor U akan ditambah dengan nilai $U_i = 2$ yang merepresentasikan kondisi dimana user menandai *item* ke- i untuk di-*blacklist*.

```

X ← MU
for j ← 1 to m do
  if  $U_j \neq 0$ 
    then  $x_j \leftarrow 0$ 
for j ← 1 to m do
  if  $x_j \neq$  among the  $N$  largest values in  $x$ 
    then  $x_j \leftarrow 0$ 
return (x)

```

Algoritma 4.2

Vektor x dihitung dengan menggunakan tiga langkah. Pertama, vektor x dihitung dengan mengalikan M dan U . Kedua, semua entri di x yang menyatakan *item* yang sudah pernah dibeli oleh user akan diset menjadi nol (0). Suatu *item* dinyatakan sudah pernah dibeli oleh user jika nilainya di matriks U adalah 1. Terakhir, di langkah ketiga, algoritma ini akan mengubah nilai semua entri yang tidak termasuk dalam N nilai terbesar di x menjadi nol (0).

Output dari algoritma ini adalah sebuah vektor x berukuran $m \times 1$ yang setiap elemen tidak nol (*nonzero*) di dalamnya menyatakan N *item* yang direkomendasikan (*top-N*). Bobot dari setiap elemen ini merepresentasikan ukuran dari *recommendation strength*. Umumnya, matriks x akan memiliki tepat N buah elemen yang nilainya bukan nol, tetapi jumlah aktual rekomendasi mungkin saja kurang dari N tergantung pada nilai k yang digunakan untuk membangun M dan jumlah *item* yang sudah pernah dibeli oleh user.

Untuk jumlah rekomendasi yang ditampilkan, berdasarkan hasil diskusi dengan beberapa orang yang sering menggunakan situs penyedia *game*, jumlah rekomendasi awal sebaiknya tidak lebih dari 5. Namun jika user ingin mendapat lebih banyak rekomendasi, telah disediakan pilihan untuk menampilkan sisa rekomendasi yang dihasilkan. Karena itu, sistem ini dirancang untuk menghasilkan 20 rekomendasi *game* (*Top-20*), namun yang pertama ditampilkan ke user hanya 5 rekomendasi teratas, sisanya akan ditampilkan jika user menginginkan (opsional).

V. ANALISIS

Rancangan sistem yang diajukan dalam *paper* ini menggunakan metode *model-based collaboration filtering* dengan memberikan rekomendasi berdasarkan kemiripan antar *game* yang ada. Rekomendasi yang dihasilkan pada umumnya akan menghasilkan *game* yang tipenya sudah familiar bagi user yang bersangkutan. Sistem ini mungkin akan jarang menghasilkan *game* yang sama sekali berbeda dengan tipe *game* yang biasa dibeli user. Namun menurut hasil diskusi dengan beberapa user hal ini tidak menjadi masalah. Hal ini dikarenakan pengguna *game* biasanya memang mencari *game* baru yang setipe dengan *game* yang biasa dia mainkan.

Keunggulan dari sistem ini jika dibandingkan dengan situs penyedia *game* yang ada saat ini antara lain :

1. Rekomendasi yang diberikan lebih akurat karena berdasarkan kemiripan antar *game*, tidak hanya berdasarkan kesamaan *genre* atau popularitas saja (*rating*).

2. Sistem ini akan lebih mampu mengakomodasi user yang menyukai *game* dari beberapa *genre* sekaligus.
3. User tidak diwajibkan untuk memberi *rating* terhadap sejumlah *item* untuk bisa memperoleh rekomendasi.
4. User dapat merevisi hasil rekomendasi dengan adanya fasilitas *blacklist* terhadap hasil rekomendasi.
5. Proses rekomendasi lebih cepat karena model hubungan antar *item* sudah dihitung sebelumnya.

Sistem ini juga memiliki beberapa kekurangan, antara lain :

1. Tidak ada mekanisme untuk membedakan user yang membeli sedikit *game* dengan user yang membeli banyak *game* sekaligus. User yang membeli atau men-*download* banyak *game* sekaligus mungkin saja hanya sekedar mencoba-coba secara acak, tidak mencari *game* tertentu, sehingga *game* yang dia beli mungkin saja sebenarnya tidak memiliki kemiripan. Sebaliknya user yang hanya membeli sedikit *game* merepresentasikan kelompok user yang memang tertarik dengan satu kelompok *game* tertentu, sehingga besar kemungkinan *game* yang dibeli memang memiliki kemiripan yang besar.
2. Pembentukan model hubungan *item-to-item* memakan waktu yang lama.
3. Kurang mampu merekomendasikan *game* yang benar-benar baru (berbeda dengan tipe *game* yang biasa dibeli user).

VI. KESIMPULAN

Pemilihan metode *model-based collaboration filtering* berdasarkan kemiripan *item-to-item* untuk menghasilkan rekomendasi dalam kasus pemilihan *game* dirasa sudah tepat dan mampu memberikan hasil yang cukup baik. Rekomendasi yang dihasilkan akan jauh lebih memuaskan dibanding rekomendasi berbasis *genre* yang umumnya disediakan oleh situs-situs *game* yang ada saat ini. Karena algoritma yang digunakan berbasis pada kemiripan *item-to-item*, maka rekomendasi yang dihasilkan pada umumnya akan berkisar pada jenis *game* yang serupa dengan *game* yang pernah dibeli atau di-*download* oleh user, dan tidak menghasilkan sesuatu

yang benar-benar di luar dugaan. Meskipun demikian, berdasarkan diskusi dengan beberapa pengguna situs *game*, umumnya mereka memang tidak mencari *game* yang benar-benar berbeda dengan tipe *game* yang selama ini mereka mainkan. Mereka cenderung lebih menginginkan rekomendasi *game* dari tipe yang memang sudah familiar, namun belum pernah mereka mainkan sebelumnya. Tidak adanya keharusan bagi user untuk memberi *rating* terlebih dahulu juga dipandang sebagai sesuatu yang memudahkan bagi user.

ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered.

Causal Productions wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template. To see the list of contributors, please refer to the top of file IEEETran.cls in the IEEE LaTeX distribution.

REFERENSI

- [1] Antonucci, Mike. New Video-Game Recommendation System? A+E Interactive. The Mercury News. MercuryNews.com. (2007)
- [2] Aranda, J., Givoni, I., Handcock, J., and Tarlow D. An Online Social Network-Based Recommendation System. Department of Computer Science, University of Toronto. (2007)
- [3] Deshpande, M. and Karypis, G. Item-Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems, 22(1):143-177. (2004)
- [4] Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. Evaluating Collaborative Filtering Recommender System. ACM Transactions on Information Systems, 22(1):5-53. (2004)
- [5] Montaner, M., Lopez, B., and Rosa, J.L.D. A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review, 19:285-330. (2003)
- [6] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. GroupLens Research Group/Army HPC Research Center. Department of Computer Science and Engineering, University of Minnesota. (2001)
- [7] Ujjin, S. and Bentley, P.J. Building a Lifestyle Recommender System. Department of Computer Science, University College London.
- [8] Vosaliz, E., and Margaritis, K.G. Analysis of Recommender System's Algorithms. <http://macedonia.uom.gr/~mans/papiria/>