

## RECOMMENDATION-BASED MENU INTERFACE UNTUK DINAMISASI DAN PERSONALISASI MENU APLIKASI

Tubagus Arief Armanda<sup>1</sup> dan Tubagus Mohammad Akhriza<sup>2</sup>

<sup>(1)</sup>STMIK Jakarta STI&K

Jl. BRI No.17, Radio Dalam, Kebayoran Baru, Jakarta Selatan 12140

<sup>(2)</sup>STMIK PPKIA Pradnya Paramita (STIMATA)

Jl. Laksda Adi Sucipto 249A, Malang, Jawa Timur

tb\_armanda@yahoo.com, akhriza@stimata.ac.id

### ABSTRAK

Menu di dalam aplikasi merupakan fasilitas penting bagi pengguna untuk dapat menemukan fitur-fitur aplikasi yang diperlukannya. Fitur sebuah aplikasi digunakan untuk menyelesaikan suatu misi tertentu. Pada aplikasi *m-banking*, misi yang diselesaikan misalnya adalah mentransfer dana ke *e-wallet* dan membayar suatu tagihan. Kecepatan pengguna menemukan fitur di dalam aplikasi menjadi salah satu kualitas menu dari aplikasi dimaksud. Pada struktur menu tradisional, pengguna seringkali harus kembali ke menu utama untuk bisa mengakses fitur-fitur lain yang berada di hirarki menu yang dalam atau di menu utama yang berbeda. Interface menu yang demikian memperlambat pengguna dalam mengakses fitur yang diperlukan, terlebih bagi pengguna baru. Artikel ini mengusulkan solusi atas permasalahan ini melalui interface menu berbasis rekomendasi, atau *recommendation-based menu interface (RBMI)* yang mampu merekomendasikan fitur-fitur Y berikutnya bagi pengguna, setelah pengguna menyelesaikan suatu fitur X. Melalui pendekatan ini, pengguna tidak perlu kembali ke menu utama untuk bisa menemukan fitur Y. Satu sesi login pengguna ke dalam aplikasi dan menggunakan beberapa fitur dianggap sebagai sebuah transaksi. Aturan asosiasi fitur-fitur ditemukan dengan metode *association rule mining*. Fitur-fitur yang digunakan tiap pengguna tentu berbeda satu dengan lainnya, maka rekomendasi fitur untuk tiap pengguna pun menjadi berbeda pula atau bersifat personal. Fitur-fitur yang direkomendasikan juga berkembang secara dinamis, mengikuti fitur-fitur yang biasa digunakan oleh pengguna. Selain itu, RBMI menyediakan akses langsung ke fitur-fitur andalan pengguna, sehingga meningkatkan efisiensi waktu pencarian dan penggunaan fitur-fitur di suatu aplikasi.

**Kata Kunci :** *Interaksi Manusia-Komputer, Interface Menu, Sistem Rekomendasi, Aturan Asosiasi*

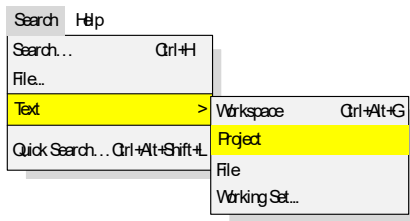
### PENDAHULUAN

Berada di era informasi, masyarakat dituntut untuk mampu menggunakan beragam aplikasi. Aplikasi dirancang dengan pengaturan menu yang sedemikian rupa sehingga memudahkan penggunaannya dalam mengakses beragam fitur di aplikasi dimaksud.

Menu aplikasi biasanya terdiri atas menu utama, submenu, sub-submenu dan seterusnya, seperti dalam Gambar 1. *Search* dan *Help* adalah item-item menu utama atau menu awal. Menu utama *Search* terdiri atas item menu *Search*, *File*, *Text* dan *Quick Search*. Menu *Text* mengandung submenu *Workspace*, *Project*, *File* dan *Working Set*. Notasi ">" dipakai untuk menunjukkan hirarki dari item-item menu. Untai hirarki "*Search>Text>Project*" bisa dibaca sebagai "menu *Search* memiliki submenu *Text* yang memiliki submenu *Project*. Item menu yang tidak memiliki

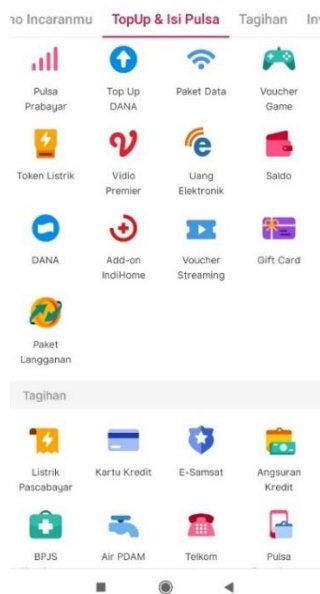
submenu lagi disebut menu akhir, misalnya item *Project* di dalam "*Search > Text >Project*" dan item *File* di dalam "*Search > File*". Pada artikel ini, menu akhir disebut fitur yaitu layanan yang disediakan oleh aplikasi untuk digunakan bagi pengguna.

Pada aplikasi berbasis piranti bergerak, menu biasanya dibuat lebih *user-friendly*, yaitu satu item menu umumnya diwakili oleh satu icon. Submenu merepresentasikan satu grup dari beberapa fitur layanan, bisa juga merepresentasikan submenu lain. Hirarki menu > submenu > ... > fitur sebagaimana dijelaskan sebelumnya, tetap berlaku. Gambar 2 adalah *screenshot* contoh menu pada aplikasi Bukalapak untuk Android.



Gambar 1. Ilustrasi Menu, Submenu dan Fitur

Pada prinsipnya, pengguna memiliki misi tertentu ketika menggunakan suatu aplikasi. Misi dianggap selesai apabila satu fitur telah dijalankan. Menurut Gambar 1, misi yang ingin dilakukan adalah untuk menemukan text di dalam sebuah project. Pada aplikasi *m-banking* atau Bukalapak, misi yang ingin dilakukan misalnya melakukan *top up* dana ke *e-wallet*. Kecepatan pengguna menemukan fitur dan menyelesaikan misi menjadi salah satu kualitas suatu antarmuka menu [1], [2].



Gambar 2. Menu di Bukalapak untuk Android

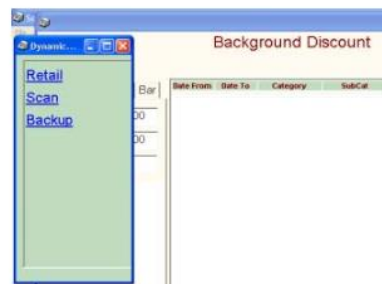
Penelitian mengenai desain menu aplikasi cukup banyak ditemukan di literatur, antara lain berfokus pada jumlah item di dalam menu [3], [4], posisi item di dalam menu [5], juga bentuk menu seperti kue (*pie*) [6] dan bunga [3] sebagai alternatif dari bentuk linear yang sudah secara umum diterapkan. Penelitian mengenai tata letak menu untuk perpindahan dari divais berbasis web dan mobile dibahas di [3]. Solusi yang ditawarkan pada penelitian-penelitian tersebut belum mampu menjawab

permasalahan ketika beberapa fitur sering diakses, namun fitur tersebut berada di dalam struktur hirarki yang cukup dalam dan berbeda menu akarnya. Hal ini menyebabkan pengguna harus kembali ke menu akar untuk masuk ke fitur berbeda. Sebagai contoh, untuk dapat mengakses fitur *top up DANA*, seorang pengguna aplikasi harus masuk ke menu *TopUp*. Untuk mengakses fitur pembayaran tagihan PLN, pengguna harus masuk ke menu *Tagihan*, padahal kedua fitur tersebut sering digunakan pengguna bersangkutan.

Sebuah artikel mengusulkan prototipe menu dinamis menggunakan pendekatan *cognitive walkthrough* (CW). Metode ini menggunakan log aktifitas untuk mengetahui waktu dan lamanya suatu jendela aplikasi dibuka, kemudian waktu jendela ini ditutup hingga waktu jendela berikutnya dibuka (Gambar 3). Kolom Win Close kode jendela yang ditutup, dan Win Open New menyebutkan kode jendela berikutnya yang dibuka. Sedangkan interval menunjukkan jeda sebelum jendela berikutnya dimaksud dibuka. Metode ini melakukan analisis statistika atas file log, kemudian menentukan item-item menu yang akan muncul di tugas berikutnya. Gambar 4 menunjukkan menu dinamis yang terdiri atas tiga item: Retail, Scan dan Backup, sebagai hasil dari analisis atas log aktifitas [7].

Opra Code	Time Close	Win Close	Time Open New	Win Open New	Interval
ADM	2007-5-9 07:24:36	w_y_rbook	2007-5-9 07:24:38	w_y_rclass	1.960000
ADM	2007-5-9 08:02:39	w_y_rclass	2007-5-9 08:02:41	w_y_rbook	1.960000
ADM	2007-5-9 08:02:44	w_y_rbook	2007-5-9 08:02:46	w_y_rclass	1.960000
ADM	2007-5-9 19:36:25	w_y_rclass	2007-5-9 19:36:26	w_y_rbook	2.970000
ADM	2007-5-9 21:45:36	w_y_rclass	2007-5-9 21:45:39	w_y_rclass	2.970000
ADM	2007-5-10 19:16:06	w_y_rclass	2007-5-10 19:16:08	w_y_rbook	1.960000

Gambar 3. Log aktifitas suatu jendela aplikasi ketika dibuka dan ditutup [7]



Gambar 4. Contoh menu dinamis yang mengambang [7]

Metode berbasis CW ini mampu meningkatkan waktu akses ke item-item yang muncul dalam jendela menu dinamis hingga 50%, namun hanya mengandalkan frekuensi diaksesnya suatu fitur saja. Akibatnya, fitur berikutnya belum tentu fitur yang memiliki peluang untuk dipilih setelah fitur yang saat ini sedang dipilih. Dari sisi desain, menu dinamis ini terus mengambang sehingga mengganggu kerja pengguna [7].

Artikel ini mengusulkan solusi atas keterbatasan metode CW dengan mengusulkan pendekatan sistem rekomendasi untuk menghasilkan menu yang dinamis dan personal bagi pengguna. Secara khusus, pendekatan ini dinamakan interface menu berbasis rekomendasi, atau *recommendation-based menu interface* (RBMI) yang dikembangkan dengan memanfaatkan konsep penambahan aturan asosiasi atau *association rule* (AR). Dengan aturan ini, fitur Y akan direkomendasikan untuk diakses oleh pengguna setelah ia selesai menggunakan fitur X. Dengan cara ini, pengguna tidak perlu kembali ke menu utama untuk mengakses Y, tapi langsung mengaksesnya dari jendela aplikasi fitur X.

## METODOLOGI PENELITIAN

### Dasar Pembentukan Aturan Asosiasi Fitur

Tugas pertama yang mendasar adalah menemukan kombinasi fitur X dan Y yang saling berhubungan, yaitu yang sering digunakan oleh pengguna. Setelah kombinasi XY ditemukan, maka Y bisa direkomendasikan untuk digunakan, setelah X digunakan.

Misalkan L adalah kejadian pengguna login ke sistem, dan  $x_i, i = 1 \dots m$  adalah semua fitur yang digunakan oleh pengguna ketika login. Kejadian login L dapat ditulis sebagai himpunan fitur yang berhasil digunakan oleh pengguna untuk menyelesaikan misi-misinya, atau:

$$L_i = \{x_1, x_2, x_3, \dots, x_m\}, \\ i = 1, 2, 3, \dots, dst.$$

Istilah “berhasil” di sini memiliki pengertian bahwa pengguna telah selesai menggunakan fitur dimaksud, yang biasanya ditandai dengan mengklik tombol Proses. *Timestamp* dari penggunaan tiap fitur tidak

diperhatikan, karena pada akhirnya pengguna akan menyelesaikan semua misinya, dan tidak peduli pada urutannya. Melalui pendekatan ini, sistem rekomendasi dengan sendirinya akan mencari aturan fitur yang bersesuaian dengan urutan penyelesaian misi dimaksud.

Aturan asosiasi fitur ditemukan menggunakan prinsip *market basket analysis*, dimana sebuah  $L_i$  dianggap sebagai suatu transaksi pembelian [8]. Diberikan dataset D dimana tiap baris berisi sebuah fitur set, yaitu  $L_i$ ; I adalah himpunan semua fitur di D. Penemuan aturan asosiasi fitur set dari D dilakukan dengan cara :

- 1) Mencari semua frequent fitur set XY di D,  $X, Y \subset I$ , yang support-nya di D  $\geq$  batas ambang minimum support (minsup); Support XY atau  $\text{sup}(XY)$  diartikan sebagai jumlah baris rekaman yang mengandung fitur set XY.
- 2) Mencari semua aturan  $R: X \rightarrow Y$  dimana:
  - a)  $|Y| = 1, |X| \geq |Y|$ ,
  - b) Confidence atau  $\text{conf}(X \rightarrow Y) \geq$  batas ambang minimum confidence (*minconf*). Nilai  $\text{conf}(X \rightarrow Y)$  dihitung dari  $\text{sup}(XY) / \text{sup}(X)$ .

Kondisi  $|Y| = 1, |X| \geq |Y|$  memiliki pengertian bahwa sistem akan merekomendasikan sebuah fitur tunggal Y untuk X dengan panjang yang lebih dari panjang Y, karena X pada dasarnya adalah  $\{x_1, x_2, x_3, \dots, x_m\}$ . Contoh: jika terdapat aturan  $\{top\ up\ ovo\} \rightarrow \{top\ up\ dana\}$  dan  $\{top\ up\ ovo\} \rightarrow \{bayar\ pln\}$ , maka setelah menyelesaikan top up OVO, merekomendasikan fitur top up danadan pembayaran PLN kepada pengguna. Pendekatan ini efektif dan efisien waktu bagi pengguna karena pengguna dapat langsung mengakses kedua fitur yang direkomendasikan tanpa harus kembali ke menu utama.

Pendekatan aturan asosiasi sudah digunakan secara luas dalam pengembangan sistem rekomendasi, seperti sistem rekomendasi buku perpustakaan [9], rekomendasi mengenai keterampilan teknologi informasi yang paling banyak

diminta di iklan lowongan kerja [10], [11], dan rekomendasi promosi karir dosen [12]. Keutamaan pendekatan aturan asosiasi adalah pada kemampuannya dalam menyediakan rekomendasi tanpa memerlukan data eksplisit mengenai tingkatan preferensi pengguna terhadap suatu item, namun cukup pada data transaksi yang bersifat implisit saja.

### Pembentukan Aturan secara Rapid

Penemuan aturan asosiasi fitur dilakukan ketika  $D$  sudah memiliki sejumlah peristiwa login, misalnya sebanyak  $N$  kali, atau selama satu hari. Ketika terjadi sejumlah peristiwa login lagi, rekaman baru ditambahkan ke  $D$ , dan aturan asosiasi fitur dicari lagi dari seluruh rekaman yang terekam. Cara ini memungkinkan administrator aplikasi untuk bisa menemukan hubungan fitur dengan nilai *support* dan *confidence* yang sebenarnya. Cara pengolahan data fitur secara temporal sebagaimana dijelaskan, akan menyebabkan *delay* dalam pemutakhiran rekomendasi yang dihasilkan. Secara umum, *delay* ini sepertinya tidak menjadi permasalahan pokok. Pada artikel ini mengulas sekaligus menerapkan pendekatan *rapid* dan hampir *real time* dalam menemukan asosiasi fitur, yaitu menggunakan pendekatan aliran data (*data stream*) [13], [14].

Prosedur pembentukan aturan asosiasi fitur berbasis aliran data ini dijelaskan sebagai berikut

1. Input: database  $D$  berisi data login  $L_i$
2. Output: database  $T$  berisi kombinasi fitur
3. Baca  $L_i$  yang terbaru di  $D$
4. Buat kombinasi fitur dari fitur-fitur di  $L_i$ , kemudian simpan di variabel KOM
5. Untuk tiap kombinasi  $\{X, Y\}$  di KOM
  - a. Jika  $\{X, Y\}$  baru di  $T$ ,  $support = 1$
  - b. Jika  $\{X, Y\}$  ada di  $T$ ,  $support += 1$

Sebagai contoh, misalkan  $L_i$  yang dibaca adalah himpunan fitur {topup\_e-wallet, topup\_pulsa, bayar\_tokenlistrik}. Maka dibentuk semua kombinasi fitur yang mungkin dari fitur-fitur di dalamnya.

Kombinasi dimaksud disimpan dalam variabel temporer KOM sebagai berikut:

```
KOM = {
{topup_e-wallet}, {topup_pulsa},
{bayar_tokenlistrik},
{topup_e-wallet, topup_pulsa},
{topup_e-wallet, bayar_tokenlistrik},
{topup_pulsa, bayar_tokenlistrik},
{topup_pulsa, topup_e-wallet,
bayar_tokenlistrik}}
```

Selanjutnya, konten dari KOM digunakan untuk memperbarui basisdata  $T$ . Jika ada rekaman kombinasi baru, maka support untuk kombinasi baru dimaksud diset sama dengan satu; tapi jika merupakan kombinasi yang sudah ada, maka  $support += 1$ . Jika kemudian dibaca  $L_i$  yang baru dan berisi {topup\_pulsa, bayar\_tokenlistrik, bayar\_pdam}, maka setelah kombinasi fitur terbentuk dan disimpan di KOM, maka kombinasi dimaksud ditambahkan ke  $T$ . Dengan demikian isi  $T$  dapat dilihat di Tabel 1, dimana entri berwarna biru adalah entri baru dengan  $support = 1$  atau entri lama dengan  $support + 1$ ; selain itu adalah entri lama yang tidak ada penambahan  $support$ .

Tabel 1. Database  $T$  setelah dua  $L_i$  ditambahkan.

Fitur set	support
{topup_e-wallet}	1
{topup_pulsa}	2
{bayar_tokenlistrik}	2
{bayar_pdam}	1
{topup_e-wallet, topup_pulsa}	1
{topup_e-wallet, bayar_tokenlistrik}	1
{topup_pulsa, bayar_tokenlistrik}	2
{topup_pulsa, bayar_pdam}	1
{bayar_tokenlistrik, bayar_pdam}	1
{topup_pulsa, topup_e-wallet, bayar_tokenlistrik}	1
{topup_pulsa, bayar_tokenlistrik, bayar_pdam}	1

Menurut [13], [14], jika efisiensi waktu menjadi pertimbangan utama, maka aturan yang ditambang dapat disaring lagi sehingga panjang  $X$  dan  $Y$  maksimal satu fitur saja. Referensi ini sesuai dengan kebutuhan untuk merekomendasikan fitur-fitur tunggal  $Y$ , untuk suatu fitur  $X$  yang baru saja diselesaikan oleh pengguna. Ini

artinya, fitur set XY dengan panjang lebih dari dua fitur tidak perlu disimpan. Sebaliknya, fitur set dengan panjang satu fitur, misalnya {topup\_e-wallet} digunakan untuk menghitung metrik *confidence* dari fitur set lainnya. Tabel T menjadi referensi untuk pembentukan aturan asosiasi fitur XY. Misalnya, terdapat aturan  $X = \{\text{topup\_ pulsa}\} \rightarrow Y = \{\text{bayar\_ tokenlistrik}\}$ , maka aturan ini memiliki  $\text{sup}(XY) = 2$  dan  $\text{conf}(XY) = \text{sup}(XY)/\text{sup}(X) = 2/2 = 100\%$ . Jika kedua nilai metrik ini memenuhi kriteria rekomendasi yang ditetapkan oleh administrator, maka Y direkomendasikan untuk diakses setelah X.

Dapat dipahami bahwa aturan asosiasi fitur dan juga rekomendasi fitur yang terbentuk antara satu pengguna dengan lainnya bersifat personal, artinya berbeda satu dengan lainnya, tergantung dari kebiasaan pengguna yang bersangkutan. Rekomendasi juga bersifat dinamis, artinya dapat bertambah sesuai dengan adanya fitur baru dan lama yang makin sering diakses oleh pengguna.

#### Desain Interface Aplikasi dan RBMI

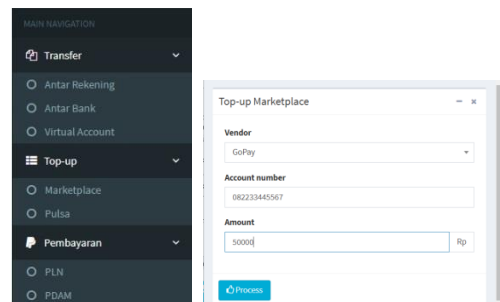
Metode yang diusulkan, yaitu *Recommendation-Based Menu Interface* – RBMI, menerapkan pembentukan aturan secara *rapid* menggunakan pendekatan data stream sebagaimana dijelaskan sebelumnya, sehingga rekomendasi dapat dihasilkan secara *hampirreal-time* juga.

Untuk menguji performa RBMI dalam menghasilkan rekomendasi fitur, sebuah aplikasi berbasis web dikembangkan (Gambar 5), dimana terlihat di area kiri terdapat tiga menu utama:

- 1) Transfer, terdiri atas tiga submenu: Antar Rekening, Antar Bank dan Virtual Account. Antar Bank memiliki beberapa jenis, yaitu BNI, Mandiri, Muamalat, sedangkan Antar Rekening dan Virtual Account tidak memiliki jenis pilihan di dalamnya.
- 2) Top Up, terdiri atas dua submenu: Marketplace dan Pulsa. Marketplace memiliki beberapa jenis yaitu Go Pay, OVO, Shopee dan Dana. Pulsa terdiri dari Telkomsel, Indosat, XL dan Three.

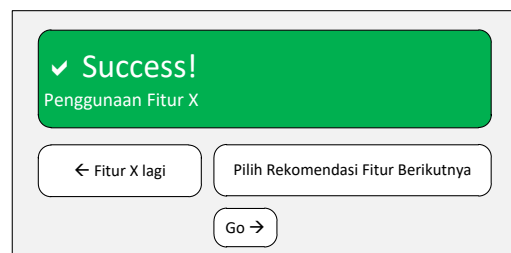
- 3) Pembayaran, terdiri atas dua submenu: PLN dan PDAM, keduanya tidak memiliki jenis pilihan lain.

Struktur hirarki menu ini cukup dalam, dan pengguna diminta untuk menjalankan beberapa fitur di menu utama yang berbeda, menurut kebiasaan mereka. Untuk menyatakan bahwa suatu fitur sudah selesai digunakan, maka pengguna harus mengklik tombol Proses, lihat Gambar 5 (kanan).



Gambar 5. (kiri) desain menu utama, (kanan) ilustrasi jendela aplikasi suatu fitur

Selama pengguna login, kode-kode fitur yang digunakan kemudian direkam ke tabel basisdata MySQL oleh program aplikasi ini. Kombinasi fitur juga langsung dibentuk dan dihitung supportnya di dalam tabel dimaksud.



Gambar 6. Desain form RBMI

Setelah pengguna logout, dan login kembali, kemudian menggunakan lagi suatu fitur X, maka rekomendasi fitur Y berikutnya yang terkait dengan X dapat dipilih oleh pengguna melalui RBMI, seperti pada Gambar 6. Rekomendasi fitur bisa lebih dari satu buah dan dapat bertambah, tergantung fitur-fitur yang biasa digunakan oleh pengguna saat ini dan di masa mendatang.

### Rancangan Eksperimen

Beberapa eksperimen dilakukan untuk menguji kedinamisan dan personalisasi menu yang dihasilkan oleh metode RBMI. Kedinamisan diukur dari perubahan rekomendasi, mengikuti penambahan fiturbaru atau pun lama yang biasa diakses oleh pengguna. Langkah pengujian kedinamisan yang dilakukan adalah sebagai berikut:

- 1) Pengguna login ke dalam aplikasi dan mengakses beberapa fitur, menyelesaikan fitur tersebut dengan menekan tombol Proses.
- 2) Pengguna logout dari aplikasi, kemudian login kembali.
- 3) Observasi rekomendasi fitur yang dihasilkan.
- 4) Ulangi langkah 1. Pengguna diminta untuk mengakses beberapa fitur lainnya.
- 5) Ulangi langkah 2 dan 3, kemudian bandingkan rekomendasi fitur yang dihasilkan.

Untuk mengukur personalisasi menu, pengguna kedua diminta untuk mengakses fitur-fitur yang berbeda dari pengguna lain. Perbedaan rekomendasi yang diberikan kepada pengguna kedua dan pertama menunjukkan adanya personalisasi menu yang dihasilkan oleh RBMI.

Selain dinamisasi dan personalisasi, eksperimen juga dilakukan untuk menghitung efisiensi waktu pencarian suatu fitur di dalam struktur hirarki menu. Efisiensi dihitung dari jumlah klik yang dibutuhkan pengguna untuk mencapai suatu fitur, yaitu dengan langkah sebagai berikut:

- 1) Hitung  $Jklik\_menu(Y)$  = jumlah klik yang diperlukan untuk sampai ke suatu fitur  $Y$ , dari menu utama sesuai dengan hirarki menu fitur dimaksud
- 2) Hitung  $Jklik\_Rek(Y)$  = jumlah klik yang diperlukan untuk mengakses fitur  $Y$  dari suatu fitur  $X$  yang merekomendasikan fitur  $Y$ .
- 3) Efisiensi klik dari fitur  $Y$  yang direkomendasi, yaitu  $ef\_klik\_rek(Y)$  dihitung dengan rumus:

$$ef\_klik\_rek(Y) = \frac{(JKlik\_menu(Y) - JKlik\_rek(Y))}{JKlik\_menu(Y)} \times 100\% \quad (1)$$

Efisiensi klik ini menyatakan usaha yang tidak perlu dilakukan, yaitu sejumlah klik, untuk mengakses suatu fitur, baik  $Y$  maupun  $X$ . Fitur  $Top\ Up > Pulsa > Indosat$  bisa diakses melalui tiga klik, yaitu klik menu utama  $Top\ Up$ , kemudian submenu  $Pulsa$ , dilanjutkan ke fitur  $Indosat$ . Jika fitur ini bisa dicapai melalui satu klik, maka terjadi penghematan sebanyak dua klik, atau sebesar  $2/3$  klik atau 67%.

### HASIL DAN PEMBAHASAN

#### Dinamisasi dan Personalisasi Rekomendasi Fitur yang Dihasilkan

Evaluasi kedinamisan dan personalisasi RBMI diukur dengan cara membandingkan rekomendasi yang dihasilkan oleh RBMI bagi dua pengguna yang berbeda, yaitu Pengguna 1 dan 2.

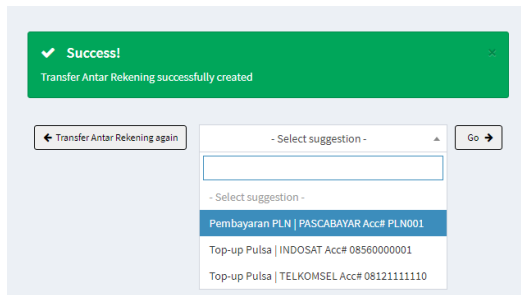
Dari hasil rekaman, fitur-fitur yang diakses oleh Pengguna 1 dan 2 diberikan di Tabel 2. Perbedaan fitur yang diakses oleh tiap pengguna menghasilkan rekomendasi fitur yang berbeda juga, seperti yang ditunjukkan dalam Gambar 7 dan 8, masing-masing untuk Pengguna 1 dan 2. Setelah melakukan transfer antar rekening, Pengguna 1 mendapatkan rekomendasi fitur berikutnya yang biasanya dipilih olehnya, yaitu: Pembayaran PLN, Top UpPulsa Indosat dan Top Up Pulsa Telkomsel. Efektifitas dari RBMI didemonstrasikan di sini.

Tiga fitur yang ditawarkan tersebut berada di dua menu utama yang berbeda yaitu menu Pembayaran dan Top Up, namun ketiganya dapat diakses secara langsung melalui *interface* RBMI yang muncul. Bukan hanya berhenti di pemilihan tiga rekomendasi ini, aplikasi RBMI juga mampu memberikan rekomendasi lebih detail, yaitu merekomendasikan vendor telekomunikasi untuk top up pulsa (Indosat, Telkomsel, XL atau Three), bahkan nomor telepon yang akan di top up juga turut direkomendasikan.

**Tabel 2.** *Fitur-fitur yang diakses oleh Pengguna 1 dan 2*

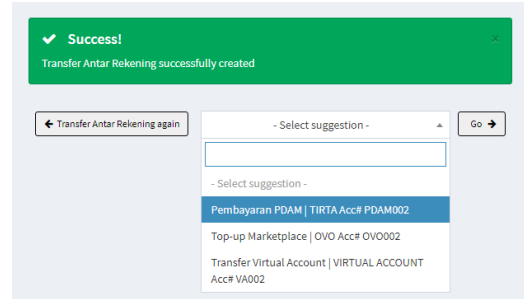
Sesi	Pengguna 1	Pengguna 2
1	Transfer > Antar Rekening  Pembayaran > PLN	Transfer >Antar Rekening  Pembayaran > PDAM
2	Transfer > Antar Rekening  Top Up > Pulsa > Indosat	Transfer >Antar Rekening  Top Up > Marketplace > OVO
3	Transfer > Antar Rekening  Top Up > Pulsa > Telkomsel	Transfer >Antar Rekening  Transfer > Virtual Account

Pada Tabel 2 di atas, Pengguna 1 dan Pengguna 2 masing-masing melakukan tiga kali sesi login, tiap sesi menyelesaikan dua buah fitur. Setiap sesi diakhiri dengan melakukan logout. Adapun hasil fitur yang direkomendasikan kepada Pengguna 1 dapat dilihat pada gambar 7.



**Gambar 7.** *Contoh fitur yang direkomendasikan kepada Pengguna 1*

Pada Gambar 8 terlihat fitur yang direkomendasikan kepada Pengguna 2. Setelah menyelesaikan transfer antar rekening, aplikasi RBMI memberikan rekomendasi Pembayaran PDAM, Topup Marketplace, atau Transfer Virtual Account.



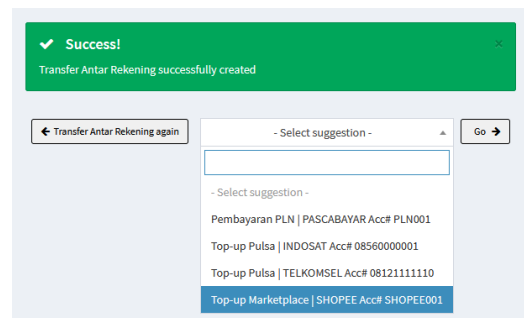
**Gambar 8.** *Contoh fitur yang direkomendasikan kepada Pengguna 2*

Daftar fitur tambahan yang diakses kedua pengguna diberikan di Tabel 3. Kedua Pengguna melakukan login kembali, kemudian masing-masing menyelesaikan dua buah fitur, diakhiri dengan logout.

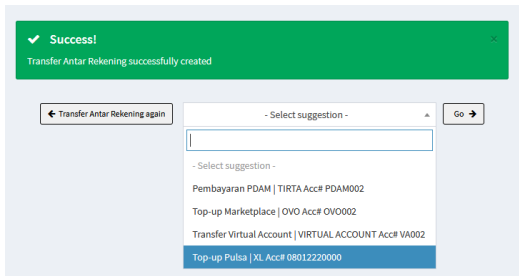
**Tabel 3.** *Fitur-fitur tambahan yang diakses oleh Pengguna 1 dan 2*

Pengguna 1	Pengguna 2
Transfer > Antar Rekening	Transfer > Antar Rekening
Top Up > Marketplace > Shopee	Top Up > Pulsa > XL

Observasi menemukan bahwa struktur menu yang dihasilkan RBMI bagi tiap pengguna sudah berubah yaitu adanya penambahan rekomendasi fitur-fitur baru, sekaligus menunjukkan bahwa menu berubah secara dinamis, mengikuti fitur-fitur baru atau pun lama yang sering diakses oleh pengguna.



**Gambar 9.** *Contoh fitur tambahan yang direkomendasikan kepada Pengguna 1*



**Gambar 10.** Contoh fitur tambahan yang direkomendasi kepada Pengguna 2

Gambar 9 dan 10 menunjukkan adanya kedinamisan ini. Jika dari fitur Transfer Antar Rekening, Pengguna 1 sebelumnya direkomendasikan untuk melanjutkan ke fitur Pembayaran PLN, TopUp Pulsa Indosat dan Top Up Pulsa Telkomsel saja, maka setelah menyelesaikan fitur tambahan, kemudian logout dan login kembali, Pengguna 1 mendapat rekomendasi fitur Top Up Marketplace. Sedangkan dengan cara yang sama, Pengguna 2 yang sebelumnya direkomendasikan fitur Pembayaran PDAM, Top Up Marketplacedan Transfer Virtual Account saja, setelah menyelesaikan fitur tambahankemudian logout dan login kembali, Pengguna 2mendapat rekomendasi fitur Top Up Pulsa XL. Jumlah rekomendasi yang berkembang menunjukkan kedinamisan menu RBMI; sedangkan fitur-fitur yang berbeda ini menunjukkan terjadinya personalisasi menu untuk tiap pengguna.

#### **Efisiensi Klik dari Fitur yang Direkomendasi**

Efisiensi waktu penemuan fitur Y yang direkomendasi, ketika fitur X selesai diakses dihitung secara langsung menggunakan rumus  $ef\_klik\_rek(Y)$  di Persamaan (1). Misalnya X adalah fitur transfer antar rekening, yang diakses melalui menu *Transfer >Antar rekening*. Apabila menggunakan menu tradisional, fitur ini ditemukan melalui dua kali klik, yaitu mengklik menu utama *Transfer*, kemudian submenu fitur *Antar Rekening*, atau  $JKlik\_Menu(X) = 2$ .

Dari menu RBMI, untuk Pengguna 1, ada empat fitur yang direkomendasikan setelah X selesai dilaksanakan, yaitu  $Y1 = Pembayaran > PLN$ ,  $Y2 = Top Up > Pulsa >$

$Indosat$ ,  $Y3 = Top Up > Pulsa > Telkomsel$  dan  $Y4 = Top Up > Marketplace > Shopee$ . Efisiensi klik untuk tiap fitur yang direkomendasi bagi Pengguna 1 diberikan di Tabel 4. Dari hasil perhitungan ini diketahui bahwa RBMI meningkatkan efisiensi penemuan suatu fitur sebesar nilai yang ditunjukkan dalam  $Ef\_Klik\_Rek$ . Untuk Pengguna 1, efisiensi sebesar 50% - 67%. Nilai ini dapat bertambah jika suatu fitur berada di posisi yang cukup dalam di dalam struktur hirarkinya. Efisiensi klik untuk tiap fitur yang direkomendasi bagi Pengguna 2 analog dengan penjelasan ini.

**Tabel 4.** Efisiensi Klik Fitur Y yang Direkomendasi setelah Fitur X pada Pengguna 1

Fitur		JKlik_ Menu(Y)	JKlik_ Rek(Y)	Ef_Klik_ Rek(Y)
X	Y			
Transfer > Antar Rekening	Y1	2	1	$(2 - 1)/2 = 50\%$
	Y2	3	1	$(3 - 1)/3 = 67\%$
	Y3	3	1	$(3 - 1)/3 = 67\%$
	Y4	3	1	$(3 - 1)/3 = 67\%$

#### **Perbandingan dengan Metode Cognitive Walkthrough**

Dibandingkan dengan metode CW, metode RBMI memiliki beberapa keunggulan. Pertama, RBMI tidak memerlukan file log yang berisi informasi mengenai jendela fitur yang dibuka dan ditutup, serta interval waktu akses jendela dimaksud; sebaliknya, RBMI secara simpel hanya membutuhkan data mengenai sekumpulan fitur (fitur set) yang selesai digunakan oleh pengguna selama satu sesi login. Kedua, Proses pembentukan rekomendasi dapat dihasilkan secara rapid dengan pendekatan penambahan aturan asosiasi dalam data stream; sedangkan



metode CW memerlukan analisis statistika, proses klasifikasi, analisis error, dan penggunaan bantuan online yang memerlukan waktu yang tidak sedikit. Selanjutnya, proses rekomendasi di RBMI mempertimbangkan support dan confidence suatu aturan, sehingga bersifat *unsupervised* dan otomatis, sedangkan di CW proses rekomendasi memerlukan supervisi administrator melalui hasil analisis statistik dan analisis lainnya. Terakhir, Interface RBMI tidak mengambang menutupi layar aplikasi seperti yang disediakan oleh metode CW, namun terintegrasi dengan area fitur dan muncul setelah fitur dimaksud selesai diproses.

## PENUTUP

Interface menu baru yaitu *recommendation-based menu interface* – RBMI dikembangkan dengan pendekatan sistem rekomendasi, sehingga mampu merekomendasikan fitur Y untuk diakses, setelah fitur X selesai digunakan oleh pengguna. Dengan adanya rekomendasi ini, pengguna tidak perlu mencari lagi fitur Y melalui menu utama, tapi langsung mengaksesnya dari halaman fitur X.

Pendekatan rekomendasi ini bersifat personal karena fitur yang direkomendasikan berbeda antara pengguna satu dengan yang lainnya, tergantung dari fitur-fitur yang biasa diakses oleh tiap pengguna. Fitur yang direkomendasikan juga bersifat dinamis yaitu dapat bertambah sesuai dengan adanya menu baru atau pun lama yang sering diakses oleh pengguna. Efisiensi penemuan fitur Y dihitung melalui jumlah klik yang dibutuhkan pada hirarki menu untuk mencapai fitur tersebut. Eksperimen menunjukkan peningkatan efisiensi penemuan fitur yang mencapai 50%-67%, tergantung dari tingkat kedalaman posisi fitur dari menu utama.

Aplikasi yang dikembangkan belum menyediakan rekomendasi fitur paling populer bagi pengguna di halaman pertamanya, dimana fasilitas ini dapat makin mempercepat pengguna menemukan fitur andalan mereka. Fasilitas ini akan ditambahkan di kegiatan penelitian mendatang.

## DAFTAR PUSTAKA

- [1] N. R. Dayama, M. Shiripour, A. Oulasvirta, E. Ivanko, and A. Karrenbauer, "Foraging-based optimization of menu systems," *Int. J. Hum. Comput. Stud.*, vol. 151, 2021.
- [2] X. Wang, "Design and evaluation of intelligent menu interface through cognitive walkthrough procedure and automated logging for management information system," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, vol. 5236 LNCS, pp. 408–418.
- [3] T. Halverson and A. J. Hornof, "The effects of semantic grouping on visual search," in *Conference on Human Factors in Computing Systems - Proceedings*, 2008.
- [4] M. D. Byrne, J. R. Anderson, S. Douglass, and M. Matessa, "Eye tracking the visual search of click-down menus," in *Conference on Human Factors in Computing Systems - Proceedings*, 1999.
- [5] M. D. Byrne, "ACT-R/PM and menu selection: Applying a cognitive architecture to HCI," *Int. J. Hum. Comput. Stud.*, vol. 55, no. 1, 2001.
- [6] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman, "An empirical comparison of Pie vs. Linear menus," in *Conference on Human Factors in Computing Systems - Proceedings*, 1988, vol. Part F130202.
- [7] X. Wang, "Using cognitive walkthrough procedure to prototype and evaluate dynamic menu interfaces: A design improvement," in *Proceedings of the 2008 12th International Conference on Computer Supported Cooperative Work in Design, CSCWD*, 2008, vol. 1.
- [8] C. Borgelt, "Find Frequent Item Sets and Association Rules with the Apriori Algorithm," 2017. .
- [9] L. D. Adistia, T. M. Akhriza, and S. Jatmiko, "Sistem Rekomendasi Buku untuk Perpustakaan Perguruan Tinggi Berbasis Association Rule," *J.*

- RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, 2019.
- [10] Latifah, T. M. Akhriza, and L. D. Adistia, "Constructing Recommendation about Skills Combinations Frequently Sought in IT Industries Based on Apriori Algorithm," *Adv. Comput. Sci. Res.*, vol. 95, Jan. 2020.
- [11] T. M. Akhriza, Y. Ma, and J. Li, "Revealing the Gap Between Skills of Students and the Evolving Skills Required by the Industry of Information and Communication Technology," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 05, pp. 675–698, 2017.
- [12] T. M. Akhriza and I. D. Mumpuni, "Quantitative class association rule-based approach to lecturer career promotion recommendation," *Int. J. Inf. Decis. Sci.*, vol. 13, no. 2, 2021.
- [13] T. M. Akhriza, Y. Ma, and J. Li, "Novel Push-Front Fibonacci Windows Model for Finding Emerging Patterns with Better Completeness and Accuracy," *ETRI J.*, vol. 40, no. 1, 2018.
- [14] T. M. Akhriza, Y. Ma, and J. Li, "A novel Fibonacci windows model for finding emerging patterns over online data stream," in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications, SSIC 2015 - Proceedings*, 2015.