

# RECOMMENDER SYSTEM DENGAN KOMBINASI APRIORI DAN CONTENT-BASED FILTERING PADA APLIKASI PEMESANAN PRODUK

Lukas Tommy<sup>1)</sup>, Chandra Kirana<sup>2)</sup>, Vivi Lindawati<sup>3)</sup>

<sup>123</sup> Teknik Informatika, STMIK Atma Luhur

<sup>123</sup> Jl. Jend. Sudirman, Selindung Baru, Pangkalpinang

Email: <sup>1</sup>lukastommy@atmaluhur.ac.id, <sup>2</sup>chandra.kirana@atmaluhur.ac.id, <sup>3</sup>1411500093@mahasiswa.atmaluhur.ac.id

## Abstrak

Sistem penjualan dan pemasaran produk sembako di Toko Menkong sampai sekarang dilakukan secara konvensional. Masalah ini menyebabkan kurang optimalnya omset yang diperoleh toko akibat dari jumlah transaksi produk yang tidak terlalu sering pada setiap harinya. Permasalahan tersebut dapat ditanggulangi dengan berbagai cara, satu di antaranya dengan membangun aplikasi berbasis Android untuk memesan produk. Tingginya mobilitas, popularitas, *open source* dan penggunaannya yang mudah menjadi bahan pertimbangan dalam pemilihan Android daripada sistem operasi dan platform yang lain. Aplikasi yang dibangun akan dilengkapi *recommender system* atau lebih dikenal sebagai sistem rekomendasi dengan memanfaatkan algoritma Apriori serta metode *content-based filtering*. Kemampuan untuk menyarankan item baru ke pengguna dengan berdasarkan kesamaan karakteristik dari item yang sebelumnya disukai pengguna merupakan alasan pemilihan metode *content-based filtering*. Faktor simpel, efisien, dan mampu menangani banyak data menjadi alasan pemilihan algoritma Apriori daripada algoritma lain. Model *prototyping*, metode berorientasi objek dan UML (Unified Modelling Language) merupakan model, metode, dan tools pengembangan perangkat lunak yang dimanfaatkan di paper ini. Pencarian dan pemesanan produk oleh pelanggan menjadi lebih mudah dengan aplikasi yang dibangun, sehingga kecepatan dan efisiensi transaksi meningkat. *Recommender system* aplikasi cukup akurat dalam menyarankan produk menurut kategori produk di keranjang, hobi pelanggan, dan pola pembelian, yaitu sebesar 73,33%.

**Kata Kunci:** *Recommender System, Apriori, Content-Based Filtering, M-Commerce, Android*

## 1. Pendahuluan

Pertumbuhan teknologi informasi, salah satunya internet yang semakin pesat di abad ke-21 mempengaruhi peradaban dan gaya hidup bermasyarakat. Kegiatan komunikasi, interaksi, serta transaksi dalam jarak jauh secara cepat, mudah, dan berbiaya murah menjadi mungkin berkat kehadiran internet [1].

Ponsel cerdas (*smartphone*), terutama yang berbasis Android pada saat ini bukanlah merupakan barang mewah, dimana hampir seluruh lapisan masyarakat

memilikinya. *Smartphone* ini digunakan oleh masyarakat sebagai media hiburan, komunikasi, berbisnis, bahkan untuk kegiatan jual beli barang antar para penggunanya melalui aplikasi tertentu. Android sendiri merupakan Sistem Operasi (SO) yang dikembangkan dengan berlandaskan *kernel Linux* yang bersifat *open source* dan paling banyak diimplementasikan pada perangkat cerdas di dunia [2].

Usaha perniagaan bisa diartikan sebagai salah satu jenis usaha dengan aktivitas pokok membeli lalu menyimpan komoditas di gudang, setelah itu menjualnya lagi tanpa diproses sebelumnya [3]. *Supermarket*, agen tunggal, pengecer, toko serba ada, toko kelontong, pusat perbelanjaan, plaza, atau pusat barang grosir adalah usaha yang dikategorikan ke dalam usaha perniagaan.

Satu dari berbagai tempat usaha perniagaan yang ada di Jebus, Kecamatan Paritiga, Kabupaten Bangka Barat adalah Toko Menkong, dimana ia menjual produk seperti suku cadang alat berat dan sembako. Sistem penjualan dan pemasaran produk sembako di Toko Menkong sampai sekarang dilakukan secara konvensional. Hal ini berarti pelanggan harus datang langsung ke toko untuk mencari dan membeli produk yang diinginkan, sehingga tidak praktis dalam hal waktu, biaya transportasi, dan tenaga yang dikeluarkan. Masalah ini menyebabkan kurang optimalnya omset yang diperoleh toko akibat dari jumlah transaksi produk yang tidak terlalu sering pada setiap harinya sekaligus faktor eksternal seperti persaingan bisnis.

Permasalahan tersebut dapat ditanggulangi dengan berbagai cara, satu di antaranya adalah dengan membangun aplikasi berbasis Android untuk memesan produk. Tingginya mobilitas, popularitas, *open source* dan penggunaannya yang mudah menjadi bahan pertimbangan dalam pemilihan Android daripada sistem operasi dan platform yang lain [4]. Konsumen nantinya bisa memesan produk sembako di Toko Menkong dengan menggunakan aplikasi yang dibangun, yang setelahnya produk tersebut akan dikirim oleh pegawai toko ke tempat konsumen.

Aplikasi yang dibangun akan dilengkapi *recommender system* atau lebih dikenal sebagai sistem rekomendasi dengan memanfaatkan algoritma Apriori serta metode *content-based filtering*. Kemampuan untuk menyarankan item baru ke pengguna dengan berdasarkan kesamaan karakteristik dari item yang sebelumnya disukai pengguna merupakan alasan pemilihan metode *content-based filtering* [5]. Faktor simpel, efisien, dan mampu menangani banyak data menjadi alasan pemilihan

algoritma *Apriori* daripada algoritma lain. Selain itu, pola-pola kombinasi item dapat dianalisa oleh algoritma ini [5].

Terdapat beberapa *paper* yang berhubungan dengan *recommender system* dan telah dipublikasi sebelum *paper* ini dikerjakan. Penggunaan metode *item collaborative filtering* pada perancangan sistem rekomendasi pakaian distro oleh Susanto [6]. Produk yang mungkin disukai konsumen bisa diprediksi berdasarkan histori produk seperti apa yang paling laris. Santoso dan Kusumaningrum [7] mengimplementasikan algoritma *naive bayes* dan metode *content-based filtering* pada *recommender system* untuk jual beli *online*. Produk yang disarankan cocok dengan kesukaan pengguna berkat penerapan 2 metode ini di *recommender system*, sehingga dapat dikatakan sukses.

Sistem rekomendasi dengan algoritma *Apriori* dan *content based filtering* yang dilaksanakan oleh Badriyah dkk. [5] dimana perpaduan 2 pendekatan ini bisa menganalisa pola-pola kombinasi *item*. Dengan berlandaskan transaksi jual beli yang dulunya dilakukan konsumen di *e-commerce*, sistem dapat memberikan saran produk. Produk yang disarankan bagi masing-masing pengguna anggota yang aktif dan terdaftar bisa diperlihatkan. Di tiap-tiap *detail* produk yang ditampilkan, bisa disarankan produk berbeda menurut persentase kemunculan milik produk berbeda dari pengguna yang saat itu menunjukkan *detail* produk. Penetapan nilai *confidence* dan *support* merupakan nilai yang berpengaruh pada banyaknya saran produk yang ditunjukkan. Semakin kecil nilai *confidence* dan *support* yang ditetapkan, semakin banyak rekomendasi yang ditunjukkan. Semakin besar nilai *confidence* dan *support* yang ditetapkan, semakin sedikit rekomendasi yang ditunjukkan.

Sarkaleh et al. [8] membangun *recommender system* untuk memandu wisata dengan *platform web*. Metode *collaborative filtering* dan *personalization*, serta algoritma *K-means clustering* digunakan dalam *paper* tersebut. Informasi jasa seperti hiburan, penginapan, biro perjalanan, toko-toko, serta restoran bisa disiapkan oleh aplikasi yang dibangun. Yanti dkk. [9] melakukan publikasi mengenai rekomendasi *hardware game* pada Toko Pelita Abadi Games dengan penerapan *collaborative filtering*. Angka *mean absolut error* yang dihasilkan pada eksperimen dengan metode ini termasuk akurat, yaitu sebesar 0,739.

Badriyah dkk. [10] menciptakan *recommender system hybrid* yang menerapkan pendekatan *collaborative filtering* dan *content-based filtering* pada *e-commerce*. Ia nantinya mengkalkulasi kesamaan profil pengguna dan gambaran produk. Hasil *tag* yang diciptakan secara otomatis dicampur dengan profil pengguna oleh sistem yang dibangun untuk memperoleh saran yang berkaitan. Pada hasil pengujian, ditemukan bahwa saran mempunyai kesamaan dengan gambaran produk dan kesukaan profil pengguna dengan rata-rata nilai keakuratan sebesar 67,5% dan nilai *recall* sebesar 71,47%. Penelitian ini memanfaatkan metode *Text Mining TF-IDF (term frequency-inverse document frequency)* dalam menciptakan *tag* secara otomatis sesuai gambaran produk.

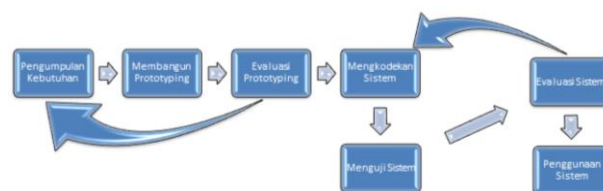
Armando [11] menerapkan sistem rekomendasi yang menggunakan *content-based filtering* dan algoritma *Apriori* pada *e-commerce*. Konsumen bisa dengan cepat mendapatkan *item* yang diinginkan dengan memanfaatkan *feature recommender system* pada *e-commerce*.

Diharapkan dengan kehadiran aplikasi yang akan dibangun bisa membuat pangsa pasar Toko Menkong menjadi lebih luas. Selain itu juga diharapkan dapat mempercepat, mempermudah, dan menjadikan transaksi produk lebih efisien sekaligus efektif, sehingga dapat memaksimalkan omset toko.

## 2. Metode

### 2.1. Model Pengembangan Perangkat Lunak

Pada penelitian ini, model *prototyping* dipilih sebagai model pengembangan perangkat lunak, yaitu satu dari sekian metode siklus hidup sistem yang berlandaskan pada model konsep bergerak guna membangun model ke bentuk sistem final [12]. Fase-fase pada model *prototyping* dapat dilihat pada Gambar 1.



Gambar 1. Fase-fase model *prototyping*.

#### A. Pengumpulan Kebutuhan

Sistem penjualan sekaligus dan pemasaran produk di Toko Menkong dilakukan secara konvensional, dimana pembeli harus datang langsung ke toko untuk mencari dan membeli produk yang ada. Hal ini menyebabkan toko tidak mampu bersaing dengan *minimarket* sekitar dimana memiliki fasilitas dan modal yang lebih baik, mengakibatkan tidak maksimalnya pendapatan yang diperoleh. Salah satu upaya agar Toko Menkong mampu bersaing dengan pesaingnya adalah dengan membuat aplikasi pemesanan produk berbasis Android. Aplikasi ini nantinya dapat digunakan untuk mencari dan memesan produk tanpa harus datang ke Toko Menkong, kemudian produk tersebut akan diantarkan oleh kurir ke lokasi pelanggan.

#### B. Membangun *Prototyping*

Pada fase ini akan dilaksanakan desain tampilan *layout* sekaligus fitur-fitur pada aplikasi. Di fase ini juga akan didesain *pseudocode* dari *recommender system* yang nanti akan diterapkan di aplikasi.

#### C. Evaluasi *Prototyping*

Purwa-rupa (prototipe) dari fase terdahulu kemudian dievaluasi oleh para *stakeholders* Toko Menkong (pengelola, karyawan, dan pelanggan) dalam hal antarmuka sekaligus fungsi. *Developer* aplikasi akan mengumpulkan kebutuhan lagi dengan *feedback* dari para *stakeholders* jika purwarupa belum disetujui.

#### D. Mengkodekan Sistem

Pada fase ini, rancangan *layout* dan fungsi diterapkan menjadi aplikasi dengan fungsi yang berjalan seutuhnya. Proses desain tampilan dan *coding* menggunakan *Integrated Development Environment (IDE)* Android Studio dengan bahasa pemrograman Java. *Recommender system* yang akan dibangun menggunakan algoritma *apriori* (berdasarkan skema pembelian produk oleh pelanggan pada umumnya) dan metode *content-based filtering* (berdasarkan hobi yang dipilih pelanggan saat mendaftar akun dan ciri-ciri produk di keranjang pelanggan).

#### E. Menguji Sistem

*Blackbox testing* dipilih sebagai model pengujian dalam menguji fungsi-fungsi (*input* dan *output*) yang dimiliki aplikasi [13]. Selain itu juga akan dilakukan simulasi untuk mengukur kemampuan *recommender system* dalam merekomendasikan produk berdasarkan suatu kasus .

#### F. Evaluasi Sistem

Aplikasi yang telah lolos pengujian *blackbox* dan simulasi akan diuji sekali lagi oleh para *stakeholders* apakah telah memenuhi kebutuhan sebelum dipublikasikan. *Developer* akan mengulangi fase mengkodekan sistem berbekal *feedback* yang diberikan seandainya aplikasi belum sesuai keinginan para *stakeholders*.

#### G. Penggunaan Sistem

Aplikasi yang sudah dievaluasi dan dinyatakan lulus oleh para *stakeholders* kemudian di-*hosting* pada suatu penyedia layanan *hosting*. *Android Package File (APK)* aplikasi selanjutnya diunggah pada Google Play Store agar dapat diunduh oleh calon pembeli untuk mencari dan memesan produk. Promosi APK ini dapat dilakukan menggunakan sosial media, mulut ke mulut ataupun dicetak pada spanduk toko.

### 2.2. Metode Pelaksanaan Penelitian

Metode berorientasi objek yang digambarkan menggunakan *Unified Modeling Language (UML)* dipilih sebagai metode pengembangan sistem dalam membangun aplikasi ini. Menurut Nugroho [14], pendekatan berorientasi objek yaitu sebuah upaya membangun sistem informasi dan *software* berlandaskan abstraksi objek – objek yang terdapat pada kehidupan sehari-hari. Model berorientasi objek memudahkan *software* untuk dimodifikasi dan lebih fleksibel oleh karena itu tepat untuk diterapkan di *software* yang nantinya dibangun. Salah satu contoh bahasa pemrograman yang menunjang *Object-Oriented Programming (OOP)* adalah Java yang dipakai di Android Studio.

### 2.3. Tools Pengembangan Sistem

UML merupakan *tools* pengembangan sistem yang dipakai dalam membangun aplikasi ini. Menurut Nugroho [15], UML adalah bahasa baku untuk memodelkan proses bisnis dan *software* yang biasanya dimanfaatkan pada pendekatan berorientasi objek. Diagram *sequence*,

diagram *class*, diagram *activity*, dan diagram *use case* merupakan diagram-diagram UML yang dipakai pada pembangunan aplikasi ini.

### 2.4. Algoritma Apriori

Algoritma *Apriori* adalah salah satu algoritma untuk menjalankan pencarian *frequent itemset* dengan *association rules*. Algoritma *Apriori* memakai pendekatan *level-wise search*, dimana *k-itemset* dipakai untuk mendapatkan *(k+1)-itemset*. Proses ini dilaksanakan sampai tidak ada lagi kombinasi yang dapat dibentuk [16]. Berdasarkan persamaan-persamaan mengenai *support* dan *confidence* yang ada di [16], serta ulasan di [17], didapatkan persamaan (1), (2), dan (3).

Persamaan (1) digunakan untuk menghitung nilai *support* untuk sebuah *item*.

$$Support(A) = \frac{Jumlah\ Transaksi\ mengandung\ A}{Total\ Transaksi} \quad (1)$$

Persamaan (2) digunakan untuk menghitung nilai *support* dari 2 *item*

$$Support(A, B) = \frac{Jumlah\ Transaksi\ mengandung\ A\ dan\ B}{Total\ Transaksi} \quad (2)$$

Persamaan (3) digunakan untuk menghitung nilai *confidence* dari *rule A → B*

$$confidence(A \rightarrow B) = \frac{Jumlah\ Transaksi\ mengandung\ A\ dan\ B}{Jumlah\ Transaksi\ mengandung\ A} \quad (3)$$

Algoritma *Apriori* sering dimanfaatkan di *market basket* (data transaksi). Sebagai contoh suatu *supermarket* mempunyai *market basket*, pemilik *supermarket* bisa memahami pola belanja seorang pelanggan dengan keberadaan algoritma *Apriori*. Jika seorang pelanggan belanja produk, misalkan roti, dan susu, dia memiliki peluang 50% belanja produk mentega. Skema seperti ini sangat berarti dengan keberadaan data transaksi sampai sekarang [18].

Cara kerja algoritma *Apriori* adalah sebagai berikut:

- Tetapkan *support* minimum
- Perulangan 1 : kalkulasikan *item-item* dari *support* (transaksi yang mengandung semua *item*) dengan memindai basis data bagi 1-*itemset*, sesudah 1-*itemset* diperoleh, dari 1-*itemset* apakah di atas *support* minimum, jika sudah mencukupi *support* minimum, 1-*itemset* itu nantinya menjadi pola *frequent* tinggi.
- Perulangan 2 : demi memperoleh 2-*itemset*, perlu dilaksanakan gabungan dari k-*itemset* terdahulu, setelah itu pindai basis data kembali guna mengkalkulasi *item-item* yang mengandung *support*. *itemset* yang mencukupi *support* minimum nantinya dipilih menjadi skema *frequent* tinggi dari calon
- Tentukan angka k-*itemset* dari *support* yang sudah mencukupi *support* minimum dari k-*itemset*

- e. Kerjakan proses bagi perulangan berikutnya sampai tidak ditemukan lagi *k-itemset* yang mencukupi *support* minimum.

2.5. Metode *Content-based Filtering*

Rekomendasi produk menurut hasil uraian persamaan produk yang sudah ditaksir pemakainya adalah konsep dari *content-based filtering* [19]. Profil pemakai diwujudkan oleh *content-based filtering* menurut karakteristik perwujudan sebuah produk. Misalnya karakteristik perwujudan sebuah naskah yaitu ucapan yang ada di naskah itu. Faktor yang menentukan profil pemakai ini lalu diberikan angka bobot menurut suatu parameter. Hasil rekomendasi adalah faktor yang memiliki *cosine similarity* paling tinggi.

Tahap-tahapan dari *content-based filtering* adalah:

- a. Berlandaskan sebuah *vector* unsur pembuatnya, sebuah benda dipisah-pisah.
- b. Berlandaskan bobot *vector* unsur pembuat sebuah benda, sistem nantinya menciptakan profil pemakai. Penciptaan profil pemakai dilaksanakan memanfaatkan algoritma *Term Frequency-Invers Document Frequency* (TF-IDF). Jumlah *term* pada sebuah naskah merupakan pengertian TF, sementara itu nilai IDF dapat dihitung memakai persamaan (4):

$$idf_i = \log\left(\frac{n}{df_i}\right) \tag{4}$$

*df* adalah jumlah naskah yang mempunyai *term i*, sedangkan *n* adalah jumlah seluruh naskah

- c. Berlandaskan profil pemakai itu, sistem kemudian menduga evaluasi gemar atau tidak digemarinya sebuah benda berlandaskan uraian keserupaan profil pemakai dengan *vector* unsur pembuat benda. Seandainya sistem menduga bahwa benda itu bakal digemari oleh pemakai, lalu benda itu nantinya diusulkan ke pemakai.

3. Hasil dan Analisis

3.1. Analisa Kebutuhan

Kebutuhan dari sistem yang nantinya dibangun didefinisikan melalui proses analisa kebutuhan. Kebutuhan fungsional, kebutuhan non-fungsional, dan kebutuhan pemakai adalah pembagian dari analisa kebutuhan. Hanya kebutuhan fungsional saja yang akan dijelaskan di *paper* ini dengan mempertimbangkan jumlah halaman.

A. Analisa Kebutuhan Fungsional

Aplikasi pemesanan produk yang akan dibangun memiliki kebutuhan-kebutuhan fungsional layaknya di bawah ini:

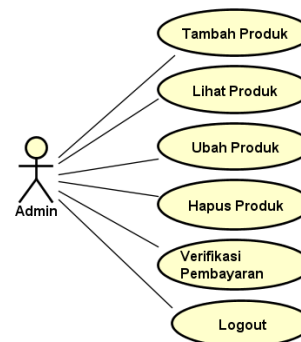
- a. Terdapat aplikasi *web* yang dibangun untuk admin, dimana dengan menggunakan aplikasi ini seorang admin dapat menambah, melihat, mengubah, ataupun menghapus produk pada basis data, melakukan verifikasi pembayaran, sekaligus *log out*. Admin harus terlebih dahulu *login* sebelum dapat

menggunakan fungsi-fungsi tersebut.

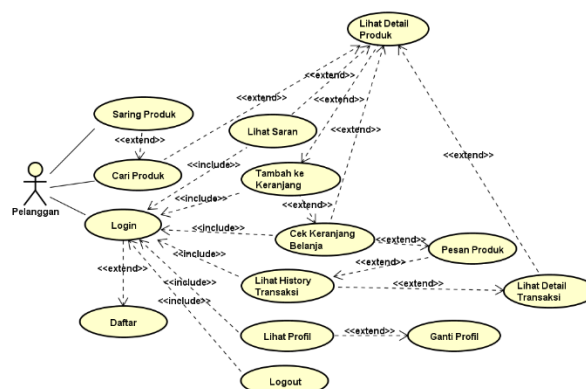
- b. Terdapat aplikasi Android yang dibangun untuk pelanggan. Dengan memanfaatkan aplikasi ini, pelanggan dapat mendaftar akun, *login*, melihat-lihat produk toko (dan penyaringan), dan melihat *detail* produk. Pelanggan hanya bisa menambahkan produk ke keranjang, mengecek dan menata keranjang, memesan produk, melihat *history* transaksi, melihat *detail* transaksi, melihat dan mengganti profil, dan memperoleh saran produk jika sebelumnya pelanggan tersebut sudah pernah mendaftar dan berhasil *login*.

B. Diagram *Use Case*

Hubungan yang berlangsung di antara *system* dan *actor* yang ada diuraikan dengan diagram *use case*. Diagram *use case* aplikasi *web* untuk admin diperlihatkan pada Gambar 2, sedangkan diagram *use case* aplikasi Android untuk pelanggan diperlihatkan pada Gambar 3:



Gambar 2. Diagram *use case* aplikasi admin.



Gambar 3. Diagram *use case* aplikasi pelanggan.

3.2. Desain Sistem

Desain yang diciptakan untuk mengembangkan aplikasi pemesanan produk, yakni representasi kelakuan aplikasi menggunakan diagram *sequence* dan diagram *activity*, desain *database* menggunakan diagram *class*. Demikian juga metode *content-based filtering* dan algoritma *Apriori* didesain menggunakan *pseudocode* dan setelah itu diimplementasikan di *recommender system* aplikasi.

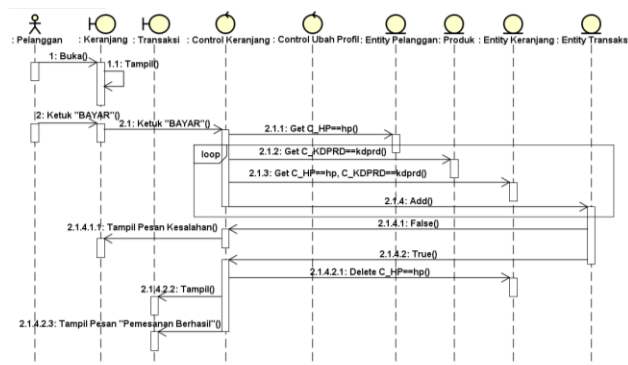
A. Diagram *Sequence*

Diagram *sequence* mendeskripsikan arus fungsi di *use case* dan menceritakan rangkaian hubungan objek yang

dikomposisi pada rangkaian periode. Tiga dari berbagai diagram *sequence* pada pembangunan aplikasi Android pemesanan produk yaitu:

1) Diagram *Sequence* Pesan Produk

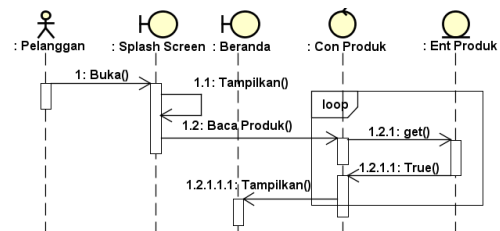
Pelanggan yang sudah *login* dan berada di halaman cek keranjang mengetuk tombol “BAYAR”. Setelah itu sistem membaca data produk yang ada di keranjang pelanggan, kemudian menambahkan transaksi itu ke *database*. Apabila terjadi kesalahan, ditampilkanlah pesan kesalahan itu, jika tidak, keranjang pelanggan akan dikosongkan dan halaman *history* transaksi ditampilkan. Diagram *sequence* pesan produk adalah seperti yang diperlihatkan Gambar 4.



Gambar 4. Diagram *sequence* pesan produk.

2) Diagram *Sequence* Cari Produk

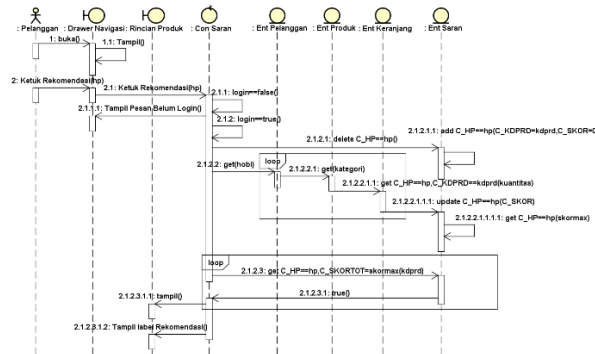
Pelanggan membuka aplikasi pemesanan produk yang dibangun, kemudian aplikasi akan menampilkan halaman *splash screen* selama 3 detik. Setelah 3 detik berlalu, sistem akan membaca semua produk yang ada di *database*, lalu menampilkannya di halaman beranda. Diagram *sequence* cari produk adalah seperti yang diperlihatkan Gambar 5.



Gambar 5. Diagram *sequence* cari produk.

3) Diagram *Sequence* Lihat Saran

Pelanggan yang sudah *login* mengetuk “Rekomendasi” pada halaman *drawer* navigasi. Setelah itu sistem menginisialisasi skor, lalu mengkalkulasikan skor yang didapat setiap produk yang tersedia di etalase berdasarkan beberapa pertimbangan. Pertimbangan ini antara lain hobi pelanggan, kategori produk yang ada di keranjang, berikut kuantitas dari produk itu dan pola pembelian pelanggan secara umum yang belanja di toko. Berdasarkan skor itu, akan dipilih produk yang direkomendasikan. Diagram *sequence* lihat saran adalah seperti yang diperlihatkan Gambar 6.



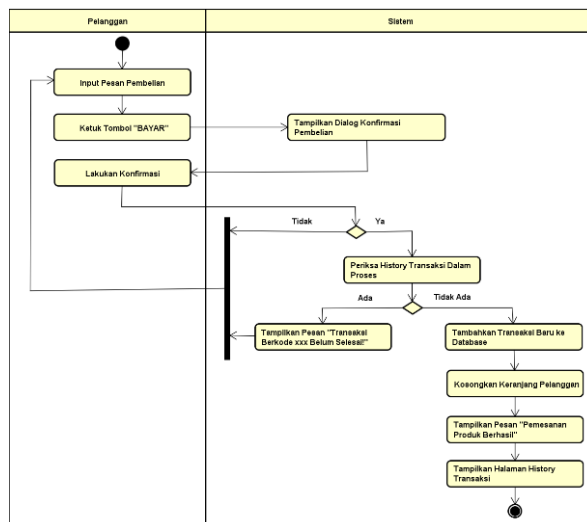
Gambar 6. Diagram *sequence* lihat saran.

B. Diagram *Activity*

Beberapa arus fungsi dari sistem yang dibangun dideskripsikan dengan diagram *activity*. Beberapa diagram *activity* pada pembangunan aplikasi Android pemesanan produk yaitu:

1) Diagram *Activity* Pesan Produk

Diagram *activity* pesan produk adalah seperti yang diperlihatkan Gambar 7.



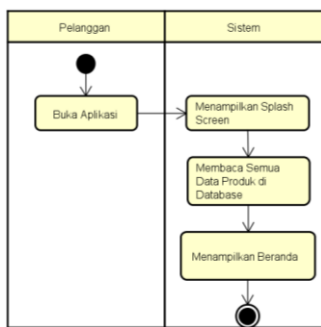
Gambar 7. Diagram *activity* pesan produk.

2) Diagram *Activity* Cari Produk

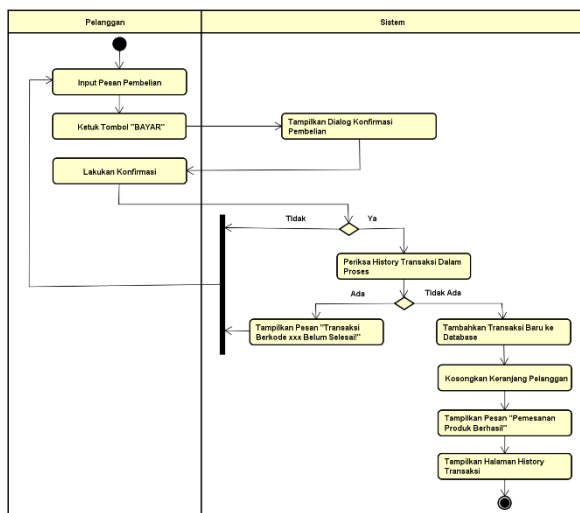
Diagram *activity* cari produk adalah seperti yang diperlihatkan Gambar 8.

3) Diagram *Activity* Lihat Saran

Diagram *activity* lihat saran adalah seperti yang diperlihatkan Gambar 9.



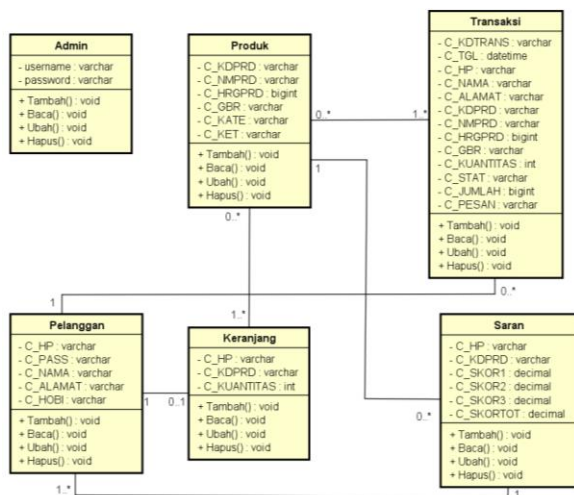
Gambar 8. Diagram activity cari produk.



Gambar 9. Diagram activity lihat saran.

C. Diagram Class

Relasi di antara kelas dalam sistem yang tersusun atas kelas, atribut, dan metode dideskripsikan menggunakan diagram class. Semua visibilitas atribut kelas yang ada bersifat *private* (nilainya tidak bisa diakses secara langsung), sedangkan metode bersifat *public*. Diagram class aplikasi adalah seperti yang diperlihatkan Gambar 10.



Gambar 10. Diagram class aplikasi.

D. Desain Algoritma

Desain algoritma di *paper* ini dicontohkan ke wujud

*pseudocode*, yakni penulisan algoritma yang memakai susunan bahasa pemrograman yang simpel. *Pseudocode* beberapa algoritma yang dipakai pada *recommender system* yang akan dibuat adalah sebagai berikut:

1) Pemberian Skor Awal

Saat pelanggan yang sudah *login* mengetuk navigasi rekomendasi di halaman *drawer* navigasi pertama kali, seluruh skor untuk semua produk yang ada di *database* untuk pelanggan tersebut akan diberi nilai 0 (nol). *Pseudocode* pemberian skor awal adalah seperti yang diperlihatkan Gambar 11.

```
Hapus semua record tabel saran dengan
C_HP=="08xx";
Tambahkan ke tabel saran semua record
C_KDPRD tabel produk, C_HP=="08xx",
C_SKOR1:=0, C_SKOR2:=0, C_SKOR3:=0,
C_SKORTOT:=0;
```

Gambar 11. *Pseudocode* pemberian skor awal.

2) Kalkulasi Skor Menurut Hobi Pelanggan

Pelanggan yang mempunyai suatu hobi biasanya condong terhadap memesan produk berkategori yang cocok dengan hobinya. Sebagai contoh, Budi yang gemar makan condong memesan produk dengan kategori makanan ringan. *Pseudocode* dari fase kalkulasi skor menurut hobi pelanggan (skor 1) yang nantinya menjadi masukan pada metode *content-based filtering* adalah seperti pada Gambar 12.

```
Cek hobi user;
Jika hobi=="makan", C_SKOR1 produk
kategori=="makanan ringan" += 0.2;
Jika hobi=="minum", C_SKOR1 produk
kategori=="minuman" += 0.2;
Jika hobi=="memasak", C_SKOR1 produk
kategori=="sembako" += 0.2;
```

Gambar 12. *Pseudocode* kalkulasi skor menurut hobi

3) Kalkulasi Skor Menurut Kategori Produk di Keranjang

Pelanggan yang mempunyai sebuah produk berkategori tertentu (X) di keranjangnya, condong memesan produk berkategori sama dengan produk itu. Misalkan Budi mempunyai 1 produk makanan ringan di keranjangnya. Dia nantinya condong untuk menambahkan produk makanan lain ke keranjangnya. Apabila telah ada cukup banyak produk makanan ringan di keranjang, maka pelanggan itu akan berhenti menambahkan produk makanan ringan ke keranjang. *Pseudocode* dari fase kalkulasi skor menurut kategori produk di keranjang (skor 2), yang akan menjadi *input* untuk metode *content-based filtering* adalah seperti yang diperlihatkan Gambar 13.

4) Kalkulasi Skor Menurut Skema Belanja Pelanggan pada Umumnya

Umumnya pelanggan yang memesan 1 atau lebih produk makanan ringan akan memesan minimal 1 produk minuman juga. Probabilitas pelanggan dalam memesan lebih banyak produk makanan ringan akan terus menurun apabila ada berbagai produk makanan ringan di

keranjangnya. Selain itu, satu orang pelanggan berpeluang besar memesan 1 minuman saja (membeli untuk dirinya sendiri) jika dia juga memesan makanan ringan pada saat itu.

Algoritma *Apriori* sejatinya hanya dapat bekerja terhadap pelanggan yang sebelumnya sudah pernah berbelanja dan histori belanjanya tercatat di *database*. Pemanfaatan skema belanja pelanggan pada umumnya membuat algoritma *Apriori* dapat bekerja bahkan untuk pelanggan yang baru pertama belanja di toko tersebut dengan menggunakan asumsi yang berlaku umum di toko sembako. *Pseudocode* dari fase kalkulasi skor menurut skema belanja pelanggan pada umumnya (skor 3), yang akan menjadi *input* untuk algoritma *Apriori* adalah seperti yang ditunjukkan Gambar 14.

```

Periksa keranjang pelanggan;
Jika produk kategori X di keranjang ==0:
    C_SKOR2 seluruh produk kategori X +=0;
Jika produk kategori X di keranjang ==1:
    C_SKOR2 seluruh produk kategori X +=0.3;
Jika produk kategori X di keranjang ==2:
    C_SKOR2 seluruh produk kategori X +=0.2;
Jika produk kategori X di keranjang >=3:
    C_SKOR2 seluruh produk kategori X +=0.1;
    
```

**Gambar 13.** *Pseudocode* kalkulasi skor menurut kategori produk di keranjang.

```

Periksa keranjang pelanggan;
Jika kuantitas produk kategori=="makanan ringan" di keranjang>=1 && kuantitas produk kategori=="minuman" di keranjang==0:
    C_SKOR3 seluruh produk kategori == "minuman" +=0.5;
Jika kuantitas produk kategori=="makanan ringan" di keranjang<=1 && kuantitas produk kategori=="minuman" di keranjang>=1:
    C_SKOR3 seluruh produk kategori == "makanan ringan" +=0.2;
Jika kuantitas produk kategori=="makanan ringan" di keranjang==2 && kuantitas produk kategori=="minuman" di keranjang>=1:
    C_SKOR3 semua barang kategori == "makanan ringan" +=0.1;
    
```

**Gambar 14.** *Pseudocode* kalkulasi skor menurut skema belanja pelanggan pada umumnya.

5) Kalkulasi Skor Total dan Saran Produk

Skor total merupakan penjumlahan dari skor 1, skor 2 dan skor 3 bagi produk tersebut, atau dengan kata lain dianggap adalah perpaduan algoritma *Apriori* dan metode *content-based filtering*. Skor total produk yang telah terdapat di keranjang tetap 0, dengan asumsi pelanggan tak akan memerlukan rekomendasi produk yang telah ada di keranjangnya. Selain itu halaman *detail* dari produk itu sudah pernah dikunjungi sebelumnya, dimana pelanggan cukup mengetuk "BELI SEKARANG" untuk menambahkan produk yang sama itu ke keranjang. Selain itu, hal ini dilaksanakan guna memprioritaskan produk lain yang belum terdapat dalam keranjang untuk

disarankan aplikasi. Skor maksimum merupakan skor total bernilai paling tinggi di antara produk yang terdapat di tabel saran. Apabila ada lebih dari 1 produk dengan skor total yang sama dengan skor maksimum, maka sebuah dari produk-produk tersebut nantinya dipilih secara acak oleh aplikasi guna disarankan ke pelanggan. *Pseudocode* dari fase kalkulasi skor total dan fase pemilihan produk yang disarankan adalah seperti yang ditunjukkan Gambar 15.

```

Update C_SKORTOT seluruh record tabel saran dengan C_HP=="08xx" dan produk tidak ada di keranjang:
    C_SKORTOT := C_SKOR1 + C_SKOR2 + C_SKOR3;
Sortir record tabel saran dengan C_HP=="08xx" berdasarkan C_SKORTOT secara descending;
SkorMaks:=C_SKORTOT record indeks ke-0;
Jika SkorMaks > 0:
    Kalkulasi jumlah record dengan C_SKORTOT ==SkorMaks;
Jika jumlah record ==1:
    Kembalikan C_KDBRG dan C_SKORTOT record indeks ke-0;
Jika jumlah record >=2:
    Acak bilangan 0 sampai dengan N-1;
    x := bilangan hasil pengacakan;
    Kembalikan C_KDBRG dan C_SKORTOT record indeks ke-x;
    
```

**Gambar 15.** *Pseudocode* kalkulasi skor total dan saran produk.

3.3. Implementasi Sistem

Implementasi sistem yang sudah dibangun pada fase desain sistem guna dioperasikan akan dilaksanakan di fase implementasi.

A. Tampilan Layar

Pada *paper* ini hanya beberapa dari seluruh tampilan layar aplikasi yang akan ditampilkan.

1) Tampilan Layar Beranda

Tampilan layar beranda dari aplikasi yang dibangun dapat dilihat pada Gambar 16.



**Gambar 16.** Tampilan layar beranda.

2) Tampilan Layar Keranjang

Tampilan layar keranjang dari aplikasi yang dibangun

dapat dilihat pada Gambar 17.



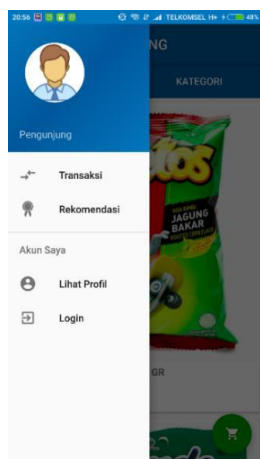
Gambar 17. Tampilan layar keranjang.

3) Tampilan Layar *Detail* Produk yang Disarankan  
 Tampilan layar *detail* produk yang disarankan dari aplikasi yang dibangun dapat dilihat pada Gambar 18.



Gambar 18. Tampilan layar *detail* produk yang disarankan.

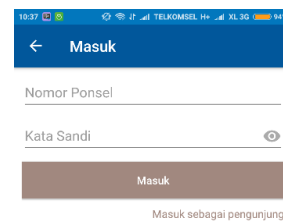
4) Tampilan Layar Navigasi *Drawer*  
 Tampilan layar navigasi *drawer* dari aplikasi yang dibangun dapat dilihat pada Gambar 19.



Gambar 19. Tampilan layar navigasi *drawer*.

5) Tampilan Layar *Login*

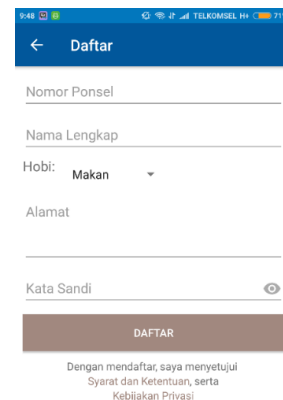
Tampilan layar *login* dari aplikasi yang dibangun dapat dilihat pada Gambar 20.



Gambar 20. Tampilan layar *login*.

6) Tampilan Layar *Daftar*

Tampilan layar *daftar* dari aplikasi yang dibangun dapat dilihat pada Gambar 21.



Gambar 21. Tampilan layar *daftar*.

7) Tampilan Layar *History* Transaksi

Tampilan layar *history* transaksi dari aplikasi yang dibangun dapat dilihat pada Gambar 22.

8) Tampilan Layar *Detail* Transaksi

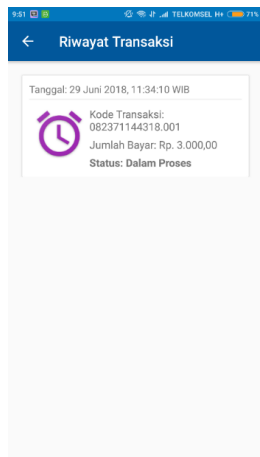
Tampilan layar *detail* transaksi dari aplikasi yang dibangun dapat dilihat pada Gambar 23.

9) Tampilan Layar *Tambah ke Keranjang*

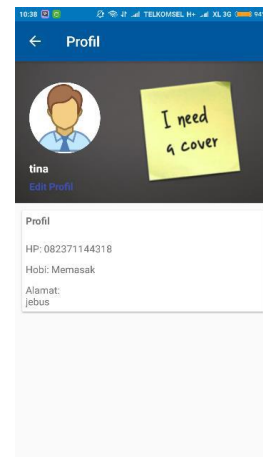
Tampilan layar *tambah ke keranjang* transaksi dari aplikasi yang dibangun dapat dilihat pada Gambar 24.

10) Tampilan Layar *Lihat Profil*

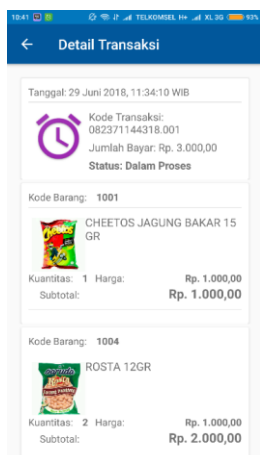
Tampilan layar *lihat profil* dari aplikasi yang dibangun dapat dilihat pada Gambar 25.



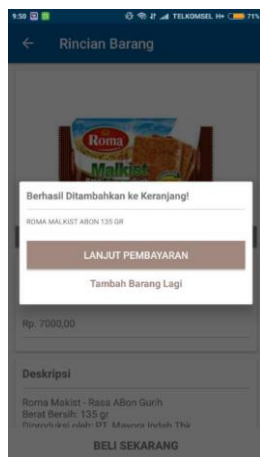
Gambar 22. Tampilan layar riwayat transaksi.



Gambar 25. Tampilan layar lihat profil.



Gambar 23. Tampilan layar detail transaksi.



Gambar 24. Tampilan layar tambah ke keranjang.

**B. Pengujian**

Pada *paper* ini, nantinya diadakan *black box testing* untuk segi fungsional dari aplikasi yang dibangun sekaligus eksperimen *recommender system* dalam menyarankan produk yang mungkin disukai oleh pelanggan.

1) *Black Box Testing*

Tabel 7 merupakan keluaran *black box testing* yang sudah dilaksanakan untuk aplikasi yang dibangun.

**Tabel 1.** Keluaran *Black Box Testing*

No	Pengujian	Hasil yang Diharapkan	Ket
1	Tap ikon launcher aplikasi Toko Menkong yang sudah dipasang pada widget ponsel Android	Menunjukkan halaman splash screen	OK
2	Menunggu 3 detik di halaman splash screen	Menunjukkan halaman beranda	OK
3	Tap salah satu kategori di halaman kategori	Menunjukkan halaman beranda dengan produk yang ditampilkan berkategori sama dengan kategori yang di-tap	OK
4	Tap sebuah produk di halaman beranda	Menunjukkan halaman detail produk yang di-tap	OK
5	Pelanggan yang sudah login meng-tap navigasi rekomendasi di halaman drawer navigasi	Menunjukkan halaman detail produk sekaligus label rekomendasi dan kalimat yang mengandung kode produk dan skornya	OK
6	Pelanggan yang sudah login meng-tap navigasi Lihat Profil di halaman drawer navigasi	Menunjukkan halaman profil	OK
7	Tap Ganti Profil pada halaman profil	Menunjukkan halaman ganti profil	OK
8	Mengganti data pelanggan dan meng-tap tombol "GANTI" di	Mengganti data pelanggan di basis data, menunjukkan kalimat ganti profil	OK

	halaman ganti profil	berhasil, menunjukkan halaman lihat profil			<i>username</i> dan sandi dengan tepat di halaman <i>login</i> admin																																										
9	Tap navigasi <i>login</i> di halaman <i>drawer</i> navigasi	Menunjukkan halaman <i>login</i>	OK	20	Admin memasukkan data produk dan mengklik tombol simpan pada halaman tambah produk	Mengentri produk ke basis data dan menunjukkan kalimat produk sukses dientri	OK																																								
10	Memasukkan nomor HP dan sandi dengan benar, meng-tap tombol "LOGIN" di halaman <i>login</i>	Menunjukkan halaman beranda dengan <i>status login</i> yaitu nomor HP itu	OK	21	Admin meng-update data produk dan mengklik tombol Ubah pada halaman <i>update</i> produk	Meng-update data produk di basis data dan menunjukkan kalimat produk sukses di-update	OK																																								
11	Meng-tap tombol Daftar di halaman <i>login</i>	Menunjukkan halaman daftar	OK	22	Admin mengklik tombol buang di baris produk pada halaman <i>update</i> buang produk	Membuang produk di basis data dan menunjukkan kalimat produk sukses dibuang	OK																																								
12	Meng-tap tombol "DAFTAR" dan seluruh data pelanggan dalam kondisi terisi di halaman daftar	Menambahkan data pelanggan ke basis data, menunjukkan kalimat daftar berhasil dan menunjukkan halaman <i>login</i>	OK	23	Admin mengklik <i>menu</i> Lap Produk di halaman admin	Menunjukkan halaman laporan produk	OK																																								
13	Pelanggan yang sudah <i>login</i> meng-tap tombol "BELI SEKARANG" di halaman <i>detail</i> produk	Menambahkan produk ke keranjang dan menunjukkan dialog pesan produk sukses ditambahkan ke keranjang	OK	24	Admin mengklik tombol tindakan verifikasi, proses, atau batalkan pada halaman verifikasi pembayaran	Mengganti status transaksi di basis data	OK																																								
14	Pelanggan yang telah <i>login</i> meng-tap ikon keranjang di halaman beranda	Menunjukkan halaman keranjang belanja pelanggan tersebut	OK	25	Admin mengklik <i>menu</i> Keluar di halaman admin	Menunjukkan halaman <i>login</i> admin	OK																																								
15	Memasukkan pesan pemesanan dan meng-tap tombol "BAYAR" di halaman keranjang belanja	Menambahkan transaksi baru ke basis data dan menunjukkan halaman <i>history</i> transaksi	OK	2) Eksperimen <i>Recommender System</i> Fase-fase eksperimen <i>recommender system</i> (simulasi) aplikasi yang dibangun yaitu seperti di bawah ini: a. Misalkan ada seorang pelanggan yang baru saja melakukan registrasi di basis data Toko Menkong dimana profilnya adalah seperti Tabel 2.																																											
16	Pelanggan yang sudah <i>login</i> meng-tap navigasi Transaksi di halaman <i>drawer</i> navigasi	Menunjukkan halaman <i>history</i> transaksi	OK	<b>Tabel 2. Profil Pelanggan Baru</b>																																											
17	Pelanggan meng-tap sebuah transaksi di halaman <i>history</i> transaksi	Menunjukkan halaman <i>detail</i> transaski	OK	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>C_HP</th> <th>C_NAMA</th> <th>C_ALAMAT</th> <th>C_HOBI</th> </tr> </thead> <tbody> <tr> <td>0821xx</td> <td>Noviyana</td> <td>Jebu Darat</td> <td>Memasak</td> </tr> </tbody> </table>				C_HP	C_NAMA	C_ALAMAT	C_HOBI	0821xx	Noviyana	Jebu Darat	Memasak																																
C_HP	C_NAMA	C_ALAMAT	C_HOBI																																												
0821xx	Noviyana	Jebu Darat	Memasak																																												
18	Pelanggan yang telah <i>login</i> meng-tap navigasi keluar di halaman <i>drawer</i> navigasi	Menunjukkan halaman <i>login</i> dan mengganti status pengunjung	OK	<b>Tabel 3. Inisialisasi Tabel Saran</b>																																											
19	Admin memasukkan	Menunjukkan halaman admin	OK	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>C_KDB</th> <th>C_SKO</th> <th>C_SKO</th> <th>C_SKO</th> <th>C_SKO</th> </tr> <tr> <th>RG</th> <th>R1</th> <th>R2</th> <th>R3</th> <th>RTOT</th> </tr> </thead> <tbody> <tr> <td>1001</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1002</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1003</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1004</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2001</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2002</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>				C_KDB	C_SKO	C_SKO	C_SKO	C_SKO	RG	R1	R2	R3	RTOT	1001	0	0	0	0	1002	0	0	0	0	1003	0	0	0	0	1004	0	0	0	0	2001	0	0	0	0	2002	0	0	0	0
C_KDB	C_SKO	C_SKO	C_SKO	C_SKO																																											
RG	R1	R2	R3	RTOT																																											
1001	0	0	0	0																																											
1002	0	0	0	0																																											
1003	0	0	0	0																																											
1004	0	0	0	0																																											
2001	0	0	0	0																																											
2002	0	0	0	0																																											
					b. Noviyana <i>login</i> , meng-tap navigasi rekomendasi di halaman <i>drawer</i> navigasi. Setelah itu dilakukan pemberian nilai awal (nol) <i>record</i> di tabel saran untuk Noviyana seperti yang diperlihatkan Tabel 3.																																										

2003	0	0	0	0
2004	0	0	0	0
3001	0	0	0	0
3002	0	0	0	0
3003	0	0	0	0
3004	0	0	0	0

- c. Noviyana gemar memasak, oleh karena itu C\_SKOR1 seluruh produk kategori sembako (kode awalan 3) ditambah 0.2. Apabila navigasi rekomendasi diketuk, nanti akan dipilih 1 dari 4 *item* yang C\_SKORTOT-nya paling tinggi (0.2) secara acak. Tabel saran bagi pelanggan Noviyana saat ini dapat dilihat di Tabel 4.

**Tabel 4.** Perhitungan Skor 1 di Tabel Saran

C_KDB RG	C_SKO R1	C_SKO R2	C_SKO R3	C_SKOR TOT
1001	0	0	0	0
1002	0	0	0	0
1003	0	0	0	0
1004	0	0	0	0
2001	0	0	0	0
2002	0	0	0	0
2003	0	0	0	0
2004	0	0	0	0
3001	0.2	0	0	0.2
3002	0.2	0	0	0.2
3003	0.2	0	0	0.2
3004	0.2	0	0	0.2

- d. Noviyana menambahkan ke keranjang Bango kecap manis botol 135 ml (kode produk 3002). Sistem mengkalkulasikan banyaknya produk di keranjang menurut kategori (tidak mempertimbangkan kuantitas). C\_SKORTOT produk yang sudah ada di keranjang dibuat 0 (agar tidak menyarankan produk yang sudah ada di keranjang). Apabila navigasi rekomendasi diketuk, sistem akan menyarankan sembako selain 3002. Tabel saran bagi Noviyana saat ini dapat dilihat di Tabel 5.

**Tabel 5.** Perhitungan Skor 2 di Tabel Saran

C_KDB RG	C_SKO R1	C_SKO R2	C_SKO R3	C_SKO RTOT
1001	0	0	0	0
1002	0	0	0	0
1003	0	0	0	0
1004	0	0	0	0
2001	0	0	0	0
2002	0	0	0	0
2003	0	0	0	0
2004	0	0	0	0
3001	0.2	0.3	0	0.5
3002	0.2	0.3	0	0
3003	0.2	0.3	0	0.5
3004	0.2	0.3	0	0.5

- e. Noviyana menambahkan minyak goreng Fortune 1 L (3001), setelah itu menambahkan Cheetos jagung

bakar 15 gr (1001) ke keranjang. Jumlah makanan ringan (kode awalan 1) di keranjang saat ini 1 buah, sehingga C\_SKOR3 seluruh produk minuman (kode awalan 2) ditambah 0.5 dengan asumsi pelanggan pada umumnya juga akan membeli minuman ketika membeli makanan ringan. Apabila rekomendasi di-*tap*, sistem nantinya memperlihatkan satu dari 4 minuman pada basis data. C\_SKOR2 sekarang menurun dari 0.3 menjadi 0.2 karena sudah ada 2 jenis produk berkategori sembako di keranjang. Tabel saran bagi Noviyana saat ini diperlihatkan di Tabel 6.

**Tabel 6.** Perhitungan Skor 2 dan Skor 3 di Tabel Saran

C_KDB RG	C_SKO R1	C_SKO R2	C_SKO R3	C_SKO RTOT
1001	0	0.3	0	0
1002	0	0.3	0	0
1003	0	0.3	0	0.3
1004	0	0.3	0	0.3
2001	0	0	0.5	0.5
2002	0	0	0.5	0.5
2003	0	0	0.5	0.5
2004	0	0	0.5	0.5
3001	0.2	0.2	0	0
3002	0.2	0.2	0	0
3003	0.2	0.2	0	0.4
3004	0.2	0.2	0	0.4

Aplikasi yang sudah dibangun setelah itu dicoba oleh 15 orang responden yang dimana hasilnya dapat dilihat di Tabel 7, dimana pertanyaan-pertanyaan pada kuesioner yaitu seperti di bawah ini:

- Aplikasi bisa dijalankan secara mudah.
- Pengalaman ketika menggunakan aplikasi ini sama seperti menggunakan aplikasi *m-commerce* sejenis seperti Tokopedia, Bukalapak, dan Shopee.
- Waktu yang dibutuhkan dalam membeli produk di Toko Menkong menjadi lebih sedikit dengan menggunakan aplikasi ini.
- Proses pemesanan produk di Toko Menkong menjadi lebih mudah dengan menggunakan aplikasi ini.
- Produk yang disarankan pada aplikasi cocok dengan keperluan Anda.

**Tabel 7.** Hasil Rekapitulasi Kuesioner

No	Nama	Soal				
		1	2	3	4	5
1	Noviyana	SS	S	SS	S	S
2	Cindy Lawrenza	S	CS	S	CS	SS
3	Yuliana	S	SS	SS	S	KS
4	Aseng	SS	S	S	S	S
5	Siti Masito	S	SS	S	CS	CS
6	Burhanudin	CS	KS	CS	KS	KS
7	Lia Risnawati	S	S	S	S	S
8	Rita Berlian	KS	CS	S	SS	KS
9	Zulkarnain	S	S	SS	S	S
10	Fatimah	SS	S	CS	CS	SS
11	Felisia	S	CS	SS	S	S
12	Tina	CS	KS	S	SS	CS

13	Rian Rendi	S	S	CS	S	S
14	Rizki Purwanto	SS	CS	S	SS	KS
15	Tomi Ali	S	S	S	S	SS

KS = Kurang Setuju, CS = Cukup Setuju, S = Setuju  
 SS = Sangat Setuju

Sebanyak 73,33% atau 11 dari 15 responden menurut persamaan (5) dan Tabel 7 setuju (selain kurang setuju/ KS) bahwa produk yang disarankan aplikasi cocok dengan keperluan mereka di waktu itu.

$$\begin{aligned} & \frac{\%Akurasi}{\text{Banyaknya Responden Setuju}} \\ & = \frac{\text{Banyaknya Responden}}{\text{Banyaknya Responden}} \\ & \times 100\% = \frac{11}{15} \times 100\% = 73,33\% \end{aligned} \quad (5)$$

#### 4. Kesimpulan

Berbagai kesimpulan yang bisa ditarik berdasarkan beberapa paparan yang sudah dijelaskan pada *paper* ini yaitu sebagai berikut:

- Proses pemesanan produk di Toko Menkong oleh pelanggan menjadi lebih cepat, mudah, dan efisien berkat aplikasi yang dibangun.
- Pelanggan yang bisa dijangkau oleh Toko Menkong menjadi lebih banyak dan dalam kawasan yang lebih luas, akibatnya jumlah transaksi yang terjadi per hari semakin banyak dan omset meningkat.
- Akurasi *recommender system* dari aplikasi yang dibangun cukup baik dalam menyarankan produk menurut hobi pelanggan, kategori produk di keranjang, dan skema belanja pelanggan pada umumnya, yakni sebesar 73,33%

#### Daftar Pustaka

- J.H. Mustakini, *Sistem Teknologi Informasi: Pendekatan Terintegrasi: Konsep Dasar, Teknologi, Aplikasi, Pengembangan, dan Pengelolaan*, 3rd ed., Andi, Yogyakarta, 2009.
- T. Arifianto, *Membuat Interface Aplikasi Android Lebih Keren dengan LWUIT*, Andi Publisher, Yogyakarta, 2011.
- Hartanto, "Analisis Pengendalian Internal Pengelolaan Pembelian Impor pada PT. XYZ", Skripsi, Fakultas Ekonomi, Universitas Esa Unggul, Jakarta, 2011.
- N.S. Harahap, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Informatika, Bandung, 2012.
- T. Badriyah, R. Fernando, & I. Syarif, "Sistem Rekomendasi Content Based Filtering Menggunakan Algoritma Apriori", *In Konferensi Nasional Sistem Informasi (KNSI) 2018*, pp. 554-559, 2018.
- H. Susanto, "Perancangan Sistem Rekomendasi Pakaian Distro dengan Menggunakan Item Collaborative Filtering (Studi Kasus : The Jungle Distro Medan)", *Pelita Informatika Budi Darma*, vol. 6, no. 3, pp. 58-62, 2014.
- Y.I. Santoso, & D.P. Kusumaningrum, Penerapan Metode Content Based Filtering untuk Sistem Rekomendasi Jual Beli Online. <http://mahasiswa.dinus.ac.id/docs/skripsi/jurnal/18699.pdf>, 2016, retrieved April 10, 2018
- M.K. Sarkaleh, M. Mahdavi, & M. Baniardalan, "Designing a Tourism Recommender System based on Location, Mobile Device and User Features in Museum", *International Journal of Managing Information Technology (IJMIT)*, vol. 4, no. 2, pp. 13-21, 2012.
- N. Yanti, R. Rahmi, & R. Ruliah, "Penerapan Algoritma Collaborative Filtering untuk Rekomendasi Games Hardware", *Jurnal Teknik Informatika dan Sistem Informasi (JUTISI)*, vol. 2, no. 1, pp. 305-314, 2013.
- T. Badriyah, E.T. Wijayanto, I. Syarif, & P. Kristalina, "A Hybrid Recommendation System for E-Commerce based on Product Description and User Profile", *In The Seventh International Conference on Innovative Computing Technology (INTECH 2017)*, pp. 95-100, 2017.
- V. Armando, "Sistem Rekomendasi Pembelian Telepon Genggam dengan Metode Content-based Filtering", Skripsi, Fakultas Teknologi Industri, Universitas Atma Jaya. Yogyakarta, 2017.
- R. McLeod Jr., *Sistem Informasi Manajemen*, Vol. 2, 7th ed., diterjemahkan oleh H. Teguh, Prenhallindo, Jakarta, 2001.
- F. Jiang, & Y. Lu, "Software Testing Model Selection Research Based on Yin-Yang Testing Theory", *In 2012 International Conference on Computer Science and Information Processing (CSIP)*, pp. 590-594, 2012.
- A. Nugroho, *Analisis dan Pemrograman Berorientasi Objek*, 2nd ed., Informatika, Bandung, 2004.
- A. Nugroho, *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*, Informatika, Bandung, 2005.
- J. Han, M. Kamber, & J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, Waltham, 2012.
- Kusrini, & E.T. Luthfi, *Algoritma Data Mining*, Penerbit ANDI, Yogyakarta, 2009.
- A. Haris, Data Mining \_ Definisi dan cara kerja Algoritma Apriori untuk pencarian association rule, <https://medium.com/@infharis/data-mining-definisi-dan-cara-kerja-algoritma-apriori-untuk-pencarian-association-rule-a44a8f864a61>, 2016, retrieved January 14 Januari, 2019.
- P.S. Adi, "Sistem Rekomendasi Nilai Mata Kuliah Menggunakan Metode Content-Based Filtering", *In Seminar Nasional Informatika 2010 (semnasIF 2010)*, pp. A-90-A94, 2010.