

Sistem Rekomendasi Pembelian Produk Kesehatan pada E-Commerce ABC berbasis Graph Database Amazon Neptune menggunakan Metode Hybrid Content-Collaborative Filtering

Made Devayani Dinda Maristha¹, Albertus Joko Santoso², Findra Kartika Sari Dewi³
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta
Jl. Babarsari 43, Kabupaten Sleman 55281, Daerah Istimewa Yogyakarta, Indonesia
Email: ¹maristhadinda@gmail.com, ²joko.santoso@uajy.ac.id, ³findra.dewi@uajy.ac.id

Abstract. *Recommendation System of Health Product Purchasing at ABC E-Commerce System based on Amazon Neptune's Graph Database using Hybrid Content-Collaborative Filtering Method.* Health products purchased by society, either in drugstores or pharmacies may vary according to their needs. ABC e-commerce is a Business to Business (B2B)-based e-commerce owned by PT XYZ. As a health product sales system from distributors to drug stores/pharmacies, they still do not have a health product purchase recommendation system yet. The recommendation system is needed to provide recommendations of health products for the customers. Amazon Neptune is implemented in this research to build a health product recommendation system. The hybrid content-collaborative filtering method is used to generate complete recommendations based on content attributes and user habits. The datasets were product data, product categories, customers, product principals, and data of products trading. This research produces a health products recommendations model at ABC e-commerce with android based using web services. The implementation can provide recommendations of health products that can be accessed in real-time by customers.

Keywords: health products, recommendation systems, graph database, Amazon Neptune, hybrid content-collaborative filtering

Abstrak. *Produk kesehatan yang dibeli masyarakat, melalui toko obat/apotek, dapat berbeda sesuai kebutuhan. E-commerce ABC berbasis Business to Business (B2B) milik PT XYZ sebagai sistem penjualan produk kesehatan dari distributor kepada toko obat/apotek belum memiliki sistem rekomendasi pembelian produk kesehatan. Sistem rekomendasi sebagai pengembangan fitur e-commerce ABC diperlukan untuk memberikan rekomendasi produk kesehatan yang sesuai dengan keadaan setiap pelanggan. Amazon Neptune sebagai graph database service yang dapat mengelola relasi dalam data yang saling terhubung, digunakan dalam penelitian untuk membangun sistem rekomendasi produk kesehatan. Metode hybrid content-collaborative filtering digunakan untuk menghasilkan rekomendasi yang lengkap berdasarkan atribut konten dan kebiasaan pengguna. Dataset yang digunakan meliputi data produk, kategori produk, pelanggan, principal, serta data jual-beli produk di e-commerce ABC. Penelitian ini menghasilkan model rekomendasi produk kesehatan yang diimplementasikan pada e-commerce ABC berbasis Android menggunakan web service. Implementasi tersebut memberikan rekomendasi produk kesehatan yang dapat diakses secara real-time oleh pelanggan pada saat menggunakan e-commerce ABC.*

Kata Kunci: produk kesehatan, sistem rekomendasi, graph database, Amazon Neptune, hybrid content-collaborative filtering

1. Pendahuluan

Produk penunjang kesehatan saat ini menjadi produk prioritas yang dicari oleh masyarakat. Produk kesehatan yang dibeli oleh masyarakat melalui toko obat/apotek dapat berbeda sesuai dengan kebutuhan mereka. Hal ini berdampak pada perbedaan jenis dan jumlah produk kesehatan yang dijual oleh toko obat/apotek sebagai penyalur produk kesehatan dari distributor. Beberapa jenis produk kesehatan dapat sangat laris jika dijual oleh toko obat/apotek

tertentu, tetapi tidak laku atau tidak dijual pada toko obat/apotek lainnya. PT XYZ sebagai salah satu perusahaan yang bergerak di bidang distribusi logistik (termasuk produk kesehatan), berperan penting dalam jual-beli produk kesehatan di masyarakat. Sistem *e-commerce* ABC berbasis *Business to Business* (B2B) yang dimiliki oleh PT XYZ sebagai sistem penjualan produk kesehatan saat ini belum memiliki sistem rekomendasi yang dapat digunakan dalam proses penjualan produk kesehatan dari distributor kepada pelanggan. Pelanggan dari *e-commerce* ABC meliputi toko obat, apotek, toko kelontong, serta bisnis lainnya yang menjual produk kesehatan. Setiap pelanggan memiliki target penjualan yang berbeda berdasarkan lokasi penjualan, sistem penjualan, kelompok pembeli, dan sebagainya. Sistem rekomendasi sebagai pengembangan fitur pada *e-commerce* ABC diperlukan untuk memberikan rekomendasi produk kesehatan yang sesuai dengan keadaan setiap pelanggan *e-commerce* ABC.

Pada penelitian ini, digunakan metode *Hybrid* agar penentuan rekomendasi dapat didasarkan pada kemiripan produk kesehatan yang ditawarkan pada *e-commerce* ABC dan juga terhadap kebiasaan pembelian yang dilakukan pelanggan. Metode ini juga sesuai dengan keadaan PT XYZ yang memiliki banyak pelanggan serta memiliki banyak jenis produk kesehatan yang dijual. Pemodelan basis data untuk pembangunan sistem rekomendasi ini menggunakan pemodelan basis data graf dengan *Amazon Neptune*. Pemodelan basis data graf atau *graph database* merupakan sistem pemodelan basis data yang sangat cocok digunakan untuk merepresentasikan dan mengeksekusi *query* pada data yang terhubung atau terkoneksi satu sama lain. Data disimpan tanpa membatasinya dengan model yang sebelumnya telah ditentukan. *Graph database* juga dapat menghasilkan gambaran atau representasi informasi yang lebih baik karena hubungan antar data dan entitas terlihat sangat jelas [1]. Implementasi sistem rekomendasi berbasis *relational database* atau basis data relasional dianggap kurang efektif dalam menggambarkan hubungan antara produk dan pelanggan. Rata-rata waktu komputasi untuk menghasilkan rekomendasi produk menggunakan *graph database* terhitung sangat cepat karena masih dalam satuan waktu *millisecond* (ms) [2]. *Amazon Neptune* merupakan salah satu *service* milik *Amazon Web Service* (AWS) yang dapat digunakan untuk membuat pemodelan basis data graf. *Amazon Neptune* merupakan *platform graph database* yang memiliki performa tinggi dan dapat memajemen *graph database* secara lengkap dan dapat dibangun dengan mudah menggunakan *Gremlin Query Language* serta *SPARQL* [3], [4]. *Query* yang digunakan pada implementasi basis data graf lebih sederhana walau sulit untuk diimplementasikan jika dibandingkan dengan *query* yang digunakan pada basis data relasional [5]. Jika sebelumnya masih menggunakan model basis data relasional, diperlukan penyeragaman pada struktur data yang digunakan dalam proses pembangunan sistem rekomendasi produk kesehatan ataupun pembangunan sistem lainnya. Hal ini perlu dilakukan agar model *graph database* hasil konversi dari model basis data relasional masih dapat memetakan struktur data dari proses bisnis yang sama dengan model basis data relasional sebelumnya [6].

Sistem rekomendasi yang akan dibangun diharapkan dapat memberikan usulan produk kesehatan sesuai dengan kebiasaan dan keadaan pelanggan, yaitu berdasarkan *history* atau riwayat pembelian produk kesehatan oleh pelanggan dan lebih spesifik berdasarkan lokasi pelanggan. Rekomendasi yang diberikan juga diperluas dengan memberikan usulan produk kesehatan lain yang berkaitan dengan produk kesehatan yang dicari oleh pelanggan. Di samping itu, pelanggan juga mendapatkan pengetahuan terkait produk kesehatan yang memiliki kemiripan dalam cara penggunaan, cara kerja, maupun manfaat produk tersebut bagi kesehatan. Hal ini dilakukan dengan membangun sistem rekomendasi produk kesehatan berdasarkan kategori dari produk kesehatan yang dicari oleh pelanggan dan *principal* pengelola produk kesehatan tersebut. Informasi produk paling laris juga akan dibangun dalam sistem rekomendasi dan diimplementasikan dalam *e-commerce* ABC. Hal ini diharapkan dapat memberikan pengetahuan kepada pelanggan terkait produk kesehatan apa saja yang laris dibeli oleh pelanggan melalui *e-commerce* ABC.

2. Tinjauan Pustaka

2.1. Sistem Rekomendasi

Sistem rekomendasi merupakan alat atau perangkat lunak yang dapat memberikan saran, usulan, anjuran kepada pengguna mengenai *item* atau sesuatu yang baik untuk mereka dapatkan atau gunakan. Istilah *item* atau sesuatu tersebut bersifat umum karena dapat mewakili banyak metode. Rekomendasi yang diberikan biasanya disesuaikan dengan jenis pengguna atau jenis kelompok pengguna tertentu. Rekomendasi yang diberikan juga dapat dipersonalisasi sehingga rekomendasi mungkin berbeda antar satu pengguna dengan pengguna lain atau dari satu kelompok ke pengguna lain [7]. Sistem rekomendasi sangat membantu pengguna yang kurang pengalaman dalam memilih dan mengevaluasi berbagai *item* yang ada. Sistem rekomendasi memiliki kemampuan untuk dapat menyajikan *item* yang tepat kepada pengguna yang tepat. Saat ini, sistem rekomendasi telah memegang peranan penting dalam berbagai industri bisnis *online* (*e-commerce*) [8]. Terdapat berbagai jenis, teknik, atau metode yang digunakan sebagai dasar pembangunan sebuah sistem rekomendasi. Tabel 1 menunjukkan jenis, teknik, atau metode pembangunan sistem rekomendasi tersebut beserta pengertian dan cara kerja masing-masing metode [9].

Tabel 1. Metode Rekomendasi

Metode Rekomendasi	Deskripsi
<i>Collaborative Filtering</i>	Menggunakan nilai kemiripan antar pengguna (<i>history</i> pengguna dalam menggunakan/menikmati suatu <i>item</i> di dalam sistem)
<i>Content-Based Filtering</i>	Menggunakan pola kemiripan antar <i>item</i> /produk yang dicari oleh pengguna di dalam sistem yang diprediksi berdasarkan profil dari pengguna
<i>Community-Based Recommender System</i>	Memberikan rekomendasi berdasarkan preferensi dari teman pengguna
<i>Demographic Recommender System</i>	Memberikan rekomendasi berdasarkan profil demografi pengguna
<i>Knowledge Based Recommender System</i>	Memberikan rekomendasi berdasarkan pengetahuan spesifik mengenai pengguna, seperti seberapa penting <i>item</i> tersebut bagi pengguna
<i>Hybrid Recommender System</i>	Mengkombinasikan dua atau lebih teknik atau metode rekomendasi

2.2. *Hybrid* (Kolaborasi Metode *Collaborative Filtering* dan *Content-Based Filtering*)

Metode *hybrid* menggabungkan teknik A dan B mencoba menggunakan keunggulan teknik A untuk memperbaiki kelemahan teknik B. Metode *hybrid* dapat diimplementasikan dengan beberapa cara, diantaranya: membuat *content-based filtering* dan *collaborative filtering* secara terpisah, lalu menggabungkan serta menambahkan keunggulan dari *content-based filtering* ke *collaborative filtering* (dan sebaliknya); atau dengan menyatukan metode atau teknik tersebut menjadi satu model. Masalah dari sistem rekomendasi *collaborative filtering* yang tidak dapat merekomendasikan *item* dari pengguna yang tidak memiliki riwayat penggunaan aplikasi akan dibantu dengan *content-based filtering* sehingga prediksi untuk *item* dapat didasarkan pada deskripsi atau fitur dari *item* yang sudah tersedia [9].

2.3. *Graph Database Amazon Neptune*

Graf merupakan representasi grafik yang melibatkan sekumpulan objek diskret. Objek diskret dilambangkan dengan simpul/*vertex* (V) dan relasi-relasi biner antar objek diskret dilambangkan dengan rusuk/sisi/*edge* (E). Graf dilambangkan dengan G, maka $G = (V_G, E_G)$. Pemodelan basis data graf atau *graph database* merupakan sistem pemodelan basis data yang sangat cocok digunakan untuk merepresentasikan dan mengeksekusi *query* pada data yang terhubung atau terkoneksi satu sama lain. *Graph database* dirancang untuk memberikan perlakuan yang sama terhadap data dan hubungan antar data tersebut. Data disimpan tanpa membatasinya dengan model yang sebelumnya telah ditentukan. *Graph database* juga dapat menghasilkan gambaran atau representasi informasi yang lebih baik karena hubungan antar data dan entitas terlihat sangat jelas. *Graph database* menyediakan evolusi dari suatu model data dan memungkinkan pembangunan model yang canggih dengan memetakannya ke domain masalah.

Model yang dihasilkan dengan basis data graf menjadi lebih sederhana dan lebih ekspresif jika dibandingkan dengan basis data relasional [1].

Amazon Neptune merupakan salah satu *service* milik *Amazon Web Service (AWS)* yang dapat digunakan untuk membuat pemodelan basis data graf. *Platform graph database* tersebut memiliki performa tinggi dan dapat mengelola *graph database* secara lengkap. *Amazon Neptune* mengelola relasi dalam data yang saling terhubung. Konsep graf dalam *Amazon Neptune* sama dengan teori graf pada umumnya, yaitu sebagai berikut: (1) **Vertex**, merupakan domain/entitas (*things, events, concepts*) pada *graph database*. Label dari *vertex* merepresentasikan *type/role* dari domain/entitas tersebut, seperti *person, product*, dan sebagainya. *Vertex* dapat memiliki properti yang merupakan atribut/metadana dari entitas, seperti *name, price, DOB, address*, dan sebagainya. (2) **Edge**, merupakan *relationship* atau relasi yang menghubungkan antar domain/entitas. Label dari *edge* merepresentasikan *tipe* dari relasi antar domain/entitas tersebut, seperti *friend, has_category, supplies*, dan sebagainya. *Edge* dapat memiliki properti yang merupakan atribut/metadana dari relasi tersebut, seperti *distance, created at, buy at*, dan sebagainya. Setiap *edge* memiliki arah dan label. *Queries* menggunakan *Gremlin Query Language* atau *SPARQL* dapat mengekspresikan *traversals* (aliran data) dari *vertex* ke *vertex* lainnya melalui *edge*. Terdapat beberapa *use case* penggunaan *graph database* terkhusus pada *graph database Amazon Neptune*, diantaranya (1) *Knowledge Graphs*, (2) *Identity Graphs*, (3) *Fraud Detection*, (4) *Recommendation Engines*, (5) *Social Networking*, (6) *Network/IT Operations*, dan (7) *Life Sciences* [3].

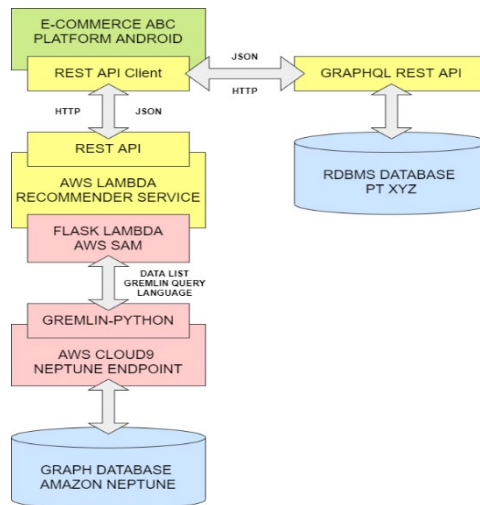
2.4. Gremlin Query Language

Gremlin Query Language merupakan salah satu *graph traversal language* yang dapat digunakan pada *graph database Amazon Neptune*. *Gremlin* dapat digunakan untuk mengambil dan memodifikasi data dalam sebuah *graph database*. Dalam pemrogramannya, *Gremlin* dapat merepresentasikan konsep dan teori graf dengan mudah karena properti dari graf disimbolkan dengan huruf yang sesuai $G = (V, E)$, yaitu *Graph (G)*, *Vertex (V)*, dan *Edge (E)*. Hal tersebut juga membuat *Gremlin* mudah untuk diimplementasikan dalam proses *traversal* (menelusuri) data yang disimpan dalam *graph database*. *Gremlin* merupakan *path-oriented query language* yang berorientasi pada jalur data dimana operator penelusuran data dirangkai bersama untuk membentuk suatu ekspresi aliran data [11].

3. Metodologi Penelitian

Pada penelitian ini, model penyimpanan data dibuat dalam bentuk graf menggunakan *graph database Amazon Neptune*. Pengembangan model *graph database* untuk sistem rekomendasi produk kesehatan dilakukan dengan metode *hybrid content-collaborative filtering* yang berfokus pada pemberian rekomendasi berdasarkan atribut konten (kategori dan *principal* pengelola produk) serta kebiasaan pengguna (*history* pembelian produk dan lokasi pengguna). Pengembangan model tersebut terdiri atas beberapa langkah sebagai berikut: (1) Pengumpulan *dataset*: *Dataset* yang digunakan adalah penjualan produk kesehatan dari tahun 2019 hingga 2020 dari salah satu cabang perusahaan PT XYZ. Data tersebut meliputi data *master* produk yang dijual, data *master* kategori dari produk yang dijual, data pelanggan, data *principal*, serta data jual-beli produk kesehatan di *e-commerce ABC*. (2) Analisis data: Langkah ini dilakukan untuk menentukan data apa saja yang digunakan dan disimpan untuk penentuan rekomendasi. (3) *Data preprocessing*: Langkah ini dilakukan untuk menghilangkan duplikasi data dan melakukan penyesuaian data dengan hasil analisis data. (4) Menyiapkan *database* pada *Amazon Neptune* untuk menampung data dalam model *graph database*. (5) Merancang model penyimpanan data pada *graph database Amazon Neptune* menggunakan *Gremlin-Python* pada *AWS Cloud9*. (6) Melakukan *import* data ke dalam *database* dari bentuk *file json* ke dalam *graph database* menggunakan model *graph database* yang sudah dibuat. (7) Melatih model penyimpanan data menggunakan pendekatan *content-based filtering* dan *collaborative filtering* untuk menghasilkan model rekomendasi produk kesehatan. (8) Mengevaluasi serta melakukan pengujian terhadap model penyimpanan data dan model rekomendasi produk kesehatan yang sudah dibangun.

Hasil rekomendasi produk kesehatan dari pembuatan model penyimpanan data dengan *graph database* tersebut diimplementasikan dalam aplikasi *mobile* berbasis *Android* sebagai pengembangan fitur pada *e-commerce ABC* milik PT XYZ. Implementasi model dilakukan dalam dua tahap, antara lain pembuatan *web service* menggunakan *Flask Lambda* untuk membaca hasil rekomendasi produk dari *graph database Amazon Neptune* dan pembuatan *Application Programming Interface (API)* menggunakan *GraphQL* untuk melengkapi detail data produk kesehatan hasil *traversal* dari *graph database*. Pada tahap akhir setelah implementasi model ke dalam sistem, dilakukan pengujian sistem rekomendasi yang dibangun untuk memastikan bahwa sistem dapat berjalan sesuai dengan yang diharapkan. Gambar 1 menunjukkan alur implementasi model untuk rekomendasi produk kesehatan ke dalam *web service*, hingga dapat digunakan pada implementasi ke dalam sistem *e-commerce ABC* berbasis *android*.



Gambar 1. Alur Implementasi Model ke dalam *Web Service*

Sistem rekomendasi produk kesehatan yang dibangun pada penelitian ini hanya menggunakan beberapa data penting dari entitas/tabel tertentu yang menjadi kunci pengembangan *graph database* untuk sistem rekomendasi produk kesehatan. Pada *graph database* entitas/tabel disebut sebagai *vertex*, sedangkan relasi antar entitas/tabel disebut sebagai *edge*. Masing-masing *vertex* dan *edge* dapat memiliki properti sebagai atribut/metadana dari entitas atau relasi yang ada. Pada *graph database*, masing-masing *vertex* dan *edge* juga harus memiliki label sebagai tanda pengenalan dan pembeda antara satu jenis *vertex/edge* dengan jenis *vertex/edge* lainnya. Tabel 2. menunjukkan *data* yang ditransformasikan ke dalam model *graph database*.

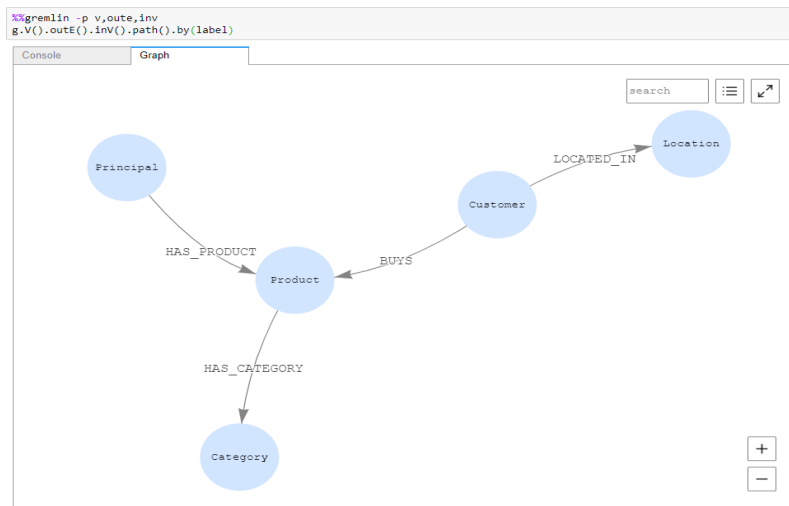
Tabel 2. Data yang ditransformasikan ke dalam Model *Graph Database*

Type	Label	Properti
vertex	Product	product_code
		product_name
vertex	Category	category_code
		category_name
vertex	Customer	customer_username
		customer_name
		ship_to_id
vertex	Location	location_name
vertex	Principal	principal_id
		principal_name
edge	HAS_CATEGORY	product_code category_code

edge	HAS_PRODUCT	principal_id
		product_code
		customer_username
edge	LOCATED_IN	ship_to_id
		location_name
		customer_address
edge	BUYS	customer_po_number
		customer_username
		ship_to_id
		product_code
		quantity_po
		creation_date

4. Hasil dan Diskusi

Model *graph database* yang digunakan untuk penyimpanan data pada *database Amazon Neptune* mengacu pada model rekomendasi produk kesehatan yang diimplementasikan. Model rekomendasi produk kesehatan yang akan diimplementasikan menggunakan pendekatan *hybrid content-collaborative filtering*. Pada model rekomendasi produk kesehatan ini, akan terbentuk empat rekomendasi produk yang dibuat berdasarkan pendekatan tersebut. Pendekatan *content-based filtering* menggunakan kategori produk dan *principal* pengelola produk sebagai dasar pemberian rekomendasi produk kesehatan. Pendekatan *collaborative filtering* menggunakan atribut *customer* sebagai dasar pemberian rekomendasi produk kesehatan. Lokasi (kota) *customer* digunakan pada pendekatan *location-based collaborative filtering* dan produk yang dibeli customer digunakan pada pendekatan *user-based collaborative filtering*. Terdapat pula tambahan model rekomendasi produk kesehatan paling laris. Gambar 2. menunjukkan model *graph database* yang sudah terbentuk pada *database Amazon Neptune*.



Gambar 2. Model Graph Database

Proses pelatihan dan evaluasi untuk model *graph database* yang sudah dibuat akan menggunakan data *customer_username* dan data *product_code* sebagai nilai masukan pada masing-masing model rekomendasi produk kesehatan yang akan diimplementasikan. Model rekomendasi untuk produk paling laris akan dilatih dan dievaluasi dengan mengakumulasi jumlah pembelian per produk. Kemudian akumulasi jumlah pembelian per produk tersebut diurutkan dari jumlah pembelian paling banyak ke jumlah pembelian paling sedikit. Pelatihan model ini menghasilkan *query* dengan *gremlin query language* yang dapat digunakan dalam proses

traversal data untuk menghasilkan rekomendasi produk kesehatan sesuai dengan pendekatan yang digunakan. Kode 1 sampai dengan Kode 5 menunjukkan *gremlin query language* yang dihasilkan dan digunakan untuk menghasilkan rekomendasi produk kesehatan.

Kode 1. Rekomendasi Produk Kesehatan *User-Based Collaborative Filtering*

```
g.V().has("Customer", "customer_username", customer_username).as("apt1")
  .out("BUYS").dedup().aggregate("pr1")
  .in("BUYS").where(neq("apt1")).dedup().as("apt2")
  .select("apt2").out("BUYS").dedup().as("pr2")
  .select("pr2").where(without("pr1")).dedup()
  .project("product_code", "product_name")
  .by("product_code").by("product_name").toList()
```

Kode 2. Rekomendasi Produk Kesehatan *Location-Based Collaborative Filtering*

```
g.V().has("Customer", "customer_username", customer_username).as("apt1")
  .out("BUYS").dedup().aggregate("pr1")
  .select("apt1").out("LOCATED_IN").dedup().as("loc1")
  .select("loc1").in("LOCATED_IN").where(neq("apt1")).dedup().as("apt2")
  .select("apt2").out("BUYS").dedup().as("pr2")
  .select("pr2").where(without("pr1")).dedup()
  .project("product_code", "product_name")
  .by("product_code").by("product_name").toList()
```

Kode 3. Rekomendasi Produk Kesehatan *Content-Based Filtering by Category*

```
g.V().has("Product", "product_code", product_code).aggregate("pr1")
  .out("HAS_CATEGORY").as("cat1")
  .select("cat1").in("HAS_CATEGORY").as("pr2")
  .select("pr2").where(without("pr1")).dedup()
  .project("product_code", "product_name")
  .by("product_code").by("product_name").toList()
```

Kode 4. Rekomendasi Produk Kesehatan *Content-Based Filtering by Principal*

```
g.V().has("Product", "product_code", product_code).aggregate("pr1")
  .in("HAS_PRODUCT").as("prin1")
  .select("prin1").out("HAS_PRODUCT").as("pr2")
  .select("pr2").where(without("pr1")).dedup()
  .project("product_code", "product_name")
  .by("product_code").by("product_name").toList()
```

Kode 5. Rekomendasi Produk Kesehatan Paling Laris

```
g.V().hasLabel("Product").inE('BUYS')
  .group().by(inV().values('product_code')).by(values('quantity_po').sum()
)
  .order(local).by(values, Order.decr).unfold()
  .project("product_code", "sum_quantity")
  .by(keys).by(values).toList()
```

Proses *traversal* untuk pendekatan *user-based collaborative filtering* dan *location-based collaborative filtering* yang ditunjukkan pada Kode 1 dan Kode 2 dimulai dengan pencarian *vertex Customer* yang memiliki variabel *customer_username*. Nilai variabel *customer_username* tersebut diambil dari *username* pelanggan yang aktif (sedang *login* dalam *e-commerce ABC*). *Traversal* dilanjutkan untuk pencarian produk yang pernah dibeli oleh pelanggan yang bersangkutan. Pada pendekatan *location-based collaborative filtering*, *traversal* dilanjutkan lagi untuk mencari lokasi dari pelanggan yang bersangkutan. Proses *traversal* pada kedua pendekatan tersebut akan menghasilkan daftar produk kesehatan yang belum pernah dibeli oleh pelanggan yang aktif berdasarkan kesamaan pembelian dengan pelanggan lainnya (pendekatan *user-based collaborative filtering*) dan kesamaan lokasi dengan pelanggan lainnya (pendekatan *location-based collaborative filtering*).

Rekomendasi produk dengan pendekatan *content-based filtering by category* dan *content-based filtering by principal* yang ditunjukkan pada Kode 3 dan Kode 4 memulai proses *traversal* dengan pencarian *vertex Product* yang memiliki variabel *product_code*. Nilai variabel *product_code* tersebut diambil dari kode produk yang dipilih untuk dilihat informasi detail produk tersebut oleh pelanggan yang aktif. Proses *traversal* dilanjutkan dengan pencarian kategori dan *principal* dari produk yang bersangkutan. Kemudian, dari kategori dan *principal* yang didapatkan, dicari produk kesehatan lainnya yang memiliki kategori dan *principal* yang sama. Kedua proses *traversal* tersebut akan menghasilkan daftar produk kesehatan sejenis berdasarkan kesamaan kategori (pendekatan *content-based filtering by category*) dan kesamaan *principal* pengelola produk (pendekatan *content-based filtering by principal*).

Proses *traversal* untuk informasi produk kesehatan paling laris yang ditunjukkan pada Kode 5 dimulai dari pencarian keseluruhan *vertex Product* yang memiliki *edge BUYS* dengan arah menuju *vertex Product*. Pencarian ini akan menghasilkan daftar produk (variabel *product_code*) beserta dengan jumlah pembelian produk tersebut (variabel *quantity_po*) yang pernah dibeli oleh pelanggan. Kemudian, hasil pencarian tersebut diurutkan berdasarkan jumlah pembelian paling banyak. Daftar produk yang sudah diurutkan tersebut merupakan daftar produk yang menjadi rekomendasi produk kesehatan paling laris untuk diinformasikan kepada pelanggan. Pada proses *traversal* dalam berbagai pendekatan rekomendasi produk kesehatan yang digunakan, terdapat beberapa *query* dasar yang digunakan pada *graph database Amazon Neptune* dengan *gremlin query language*. Penggunaan *query* tersebut dapat disesuaikan dengan model *graph database* yang sudah terbentuk setelah proses input data ke dalam *graph database*.

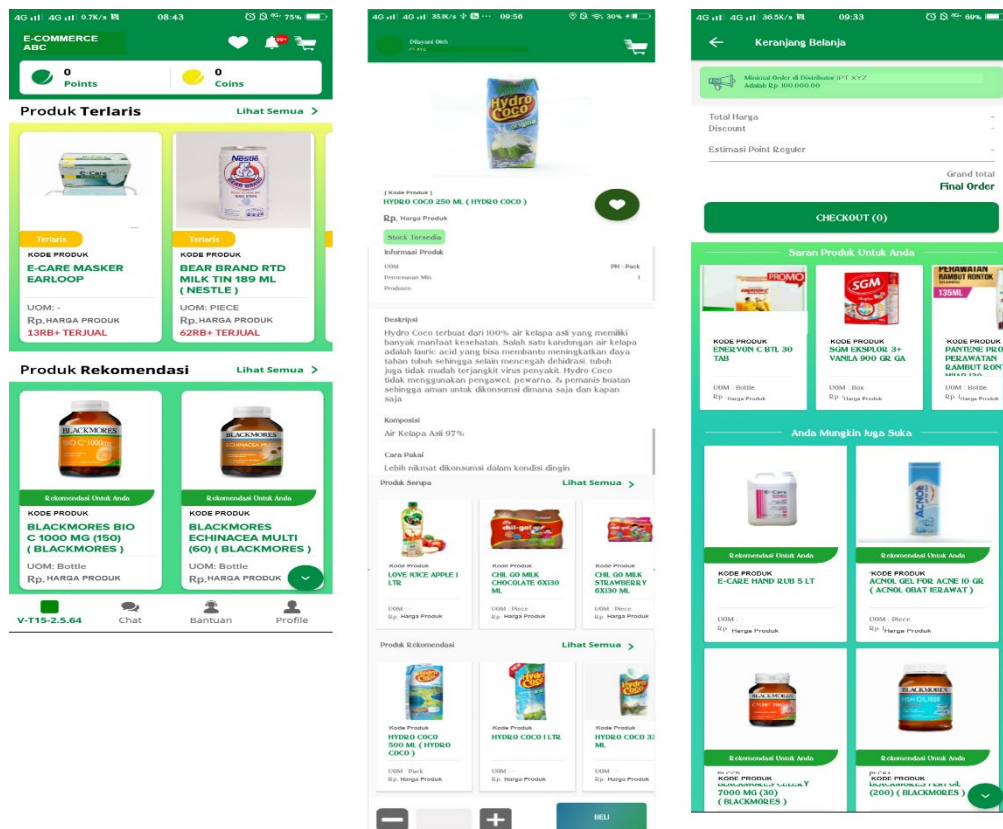
Pembuatan *web service* menggunakan *Flask Lambda* menghasilkan lima *route API* yang dapat langsung mengambil data hasil rekomendasi produk kesehatan dari proses *traversal Amazon Neptune*. Hasilnya berupa list data rekomendasi produk kesehatan dalam bentuk JSON yang berisikan variabel *product_code*, *product_name*, dan tambahan variabel *sum_quantity* (untuk produk paling laris). Pembuatan *GraphQL API* menghasilkan lima *route API* yang mengambil hasil dari *API Flask Lambda*, kemudian memberikan *response* berupa detail data hasil rekomendasi produk kesehatan yang juga dalam bentuk JSON. Detail data produk kesehatan tersebut berupa kode produk, satuan produk, harga produk, jumlah stok produk, hingga deskripsi produk. *Response* berupa detail data hasil rekomendasi produk kesehatan tersebut dibaca melalui *frontend service e-commerce ABC* berbasis *android* menggunakan library *apollo client*. Gambar 3 menunjukkan perbedaan hasil *response JSON* dari kedua *web service* yang dibuat.



Gambar 3. Perbedaan Hasil Response JSON dari Web Service Flask Lambda dan GraphQL API

Rekomendasi produk kesehatan sebagai tambahan fitur pada *e-commerce ABC* berbasis *android* diimplementasikan dalam tiga halaman tampilan *e-commerce ABC*. Ketiga halaman ialah halaman utama, halaman detail produk, dan halaman keranjang belanja. Pemberian rekomendasi daftar produk kesehatan ditampilkan secara *real-time* berdasarkan *customer_username* dari pelanggan yang sedang aktif maupun *product_code* dari produk yang sedang dicari oleh pelanggan. Daftar produk kesehatan hasil rekomendasi ditampilkan secara urut

sesuai dengan urutan abjad dari kode produk. Pada halaman utama *e-commerce ABC*, ditampilkan daftar produk kesehatan terlaris dan daftar produk rekomendasi dengan pendekatan *user-based collaborative filtering*. Pada halaman detail produk *e-commerce ABC*, ditampilkan daftar produk kesehatan hasil rekomendasi dengan pendekatan *content-based filtering by category* dan *content-based filtering by principal*. Pada halaman keranjang belanja *e-commerce ABC*, ditampilkan daftar produk kesehatan hasil rekomendasi dengan pendekatan *location-based collaborative filtering*. Data produk kesehatan yang menjadi rekomendasi untuk pelanggan ditampilkan berupa foto produk, kode produk, nama produk, UOM (satuan produk), dan harga produk. Pada produk kesehatan paling laris, terdapat tambahan data yang ditampilkan, yaitu jumlah produk terjual (*sum_quantity*). Setiap produk yang ditampilkan dapat dipilih untuk dilihat detailnya (masuk ke halaman detail produk). Kemudian, produk kesehatan tersebut dapat dipilih untuk ditambahkan ke keranjang pembelian. Gambar 4 menunjukkan pengujian sistem, yaitu tampilan rekomendasi produk kesehatan pada *e-commerce ABC* berbasis *android*.



Gambar 4. Pengujian Sistem Tampilan Rekomendasi Produk Kesehatan pada Halaman Utama, Halaman Detail Produk, dan Halaman Keranjang Belanja *E-Commerce ABC*

5. Kesimpulan dan Saran

Pembuatan model rekomendasi produk kesehatan dengan *graph database Amazon Neptune* menghasilkan daftar produk kesehatan yang menjadi rekomendasi sesuai dengan metode pendekatan *user-based collaborative filtering*, *location-based collaborative filtering*, *content-based filtering by category*, *content-based filtering by principal*, serta produk kesehatan paling laris. Model diimplementasikan sebagai tambahan fitur pada *e-commerce ABC* berbasis *android* menggunakan *web service*. Implementasi tersebut dapat memberikan rekomendasi produk kesehatan secara *real-time* berdasarkan *customer_username* milik pelanggan yang sedang aktif maupun *product_code* dari produk yang sedang dicari oleh pelanggan.

Beberapa saran dapat diberikan untuk pengembangan pada penelitian selanjutnya dalam pengembangan model menjadi lebih baik lagi. Pertama, memilih *DB Instance Size* yang lebih

besar untuk *graph database Amazon Neptune* yang digunakan. Hal ini bertujuan untuk membantu mempercepat proses komputasi yang terjadi dalam *graph database Amazon Neptune* (*write* maupun *read* data). Semakin banyak data yang disimpan dalam *graph database Amazon Neptune*, maka semakin besar *CPU* dan *RAM* yang dibutuhkan untuk memproses data tersebut agar dapat menghasilkan rekomendasi produk. Kedua, menambahkan *vertex* atau *edge* lain untuk membuat proses *traversal* yang dilakukan masih dalam satu arah aliran data. Contohnya dengan menambahkan *edge* untuk menghubungkan *vertex Location* dan *vertex Product* agar dengan mudah mengetahui rekomendasi produk pada lokasi tertentu (sebelumnya diperlukan *preprocessing* data untuk penentuan produk yang sering dibeli pada lokasi tertentu, sehingga tidak diperlukan proses *traversal* lagi ke *vertex Customer*).

Referensi

- [1] I. Robinson, J. Webber, dan E. Eifrem, *Graph Databases 2nd Edition Compliments of neo4j*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2015.
- [2] S. Marina, "Rancang Bangun Restful Web Service Pada Sistem Rekomendasi E-Commerce Berbasis Graf Neo4j dengan Metode Collaborative Filtering (Studi Kasus: Forbento)," Tugas Akhir, Dept. S. Inf., Inst. Teknologi Sepuluh Nopember, Surabaya, 2017.
- [3] Anonim, "Amazon Neptune - Fast, Reliable Graph Database built for the cloud," *Amazon*. [Daring]. Tersedia: <https://aws.amazon.com/neptune/>
- [4] Yuhanna, N., Leganza, G., & Weber, "The Forrester Wave™ Graph Data Platforms, Q4 2020," *Forrester*, 2020. [Daring]. Tersedia: <https://www.forrester.com/report/The-Forrester-Wave-Graph-Data-Platforms-Q4-2020/RES161455#>
- [5] D. E. Wibowo dan R. Munir, "Sistem Rekomendasi Jual Beli Barang dengan Memanfaatkan Metode Collaborative Filtering dan Basis Data Graf (Studi Kasus: Bukalapak.com)," Tugas Akhir, Sch. of Elect. Eng. and Inf., Inst. Technology of Bandung, 2013.
- [6] M. Sholeh, R. Y. Rachmawati, dan E. Susanti, "Pemodelan Basis data Graph dengan Neo4j (Studi Kasus : Basis Data Sistem Informasi Penjualan pada UMKM)," *Jurnal Teknologi Informasi dan Terapan*, vol. 7, no. 1, pp. 25–32, Jun. 2020.
- [7] H. Q. Cung dan M. Jedidi, "Implementing a Recommender System with Graph Database," *University of Freiburg-e Business*, 2014. [Daring]. Tersedia: https://diuf.unifr.ch/main/is/student-projects/thesis/implementing_recommender-system-graph-database.
- [8] F. M. Puri, Kusriani, dan E. T. Luthfi, "Tinjauan Pustaka Sistematis - Sistem Rekomendasi Menggunakan Collaborative Filtering," *Jurnal Technoscienza*, vol. 5, no. 1, pp. 49-58, Okt. 2020.
- [9] F. Ricci, L. Rokach, dan B. Shapira, *Recommender Systems Handbook*. Spring Street, NY, USA: Springer Science + Business Media, 2010.
- [10] K. R. Lawrence, "Practical Gremlin - An Apache TinkerPop Tutorial. Version 283-preview, October 11th 2020," *GitHub*, 2020. [Daring]. Tersedia: <https://github.com/krlawrence/graph>