

Aplikasi Graf pada Simple Digital Fraud Detection

Zidane Firzatullah - 13521163¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13521163@mahasiswa.itb.ac.id

Abstract—Fraud detection menjadi sangat penting untuk diimplementasikan karena banyaknya cara untuk melakukan frauds dengan maraknya system pembayaran digital saat ini. Graf memberikan metode fraud detection yang efisien sehingga waktu eksekusi kueri data menjadi sangat cepat relatif terhadap data dalam basis data lain.

Keywords— Simple Fraud Detection, Real Time, Graph

I. PENDAHULUAN

Di zaman yang serba digital ini, pembayaran secara digital sudah sangat marak untuk dilakukan, dari UMKM sampai toko ritel raksasa sudah hampir pasti menyediakan setidaknya satu metode pembayaran digital. Dompet digital, seperti Gopay, Ovo, Dana, dan lain-lain, kartu kredit, kartu debit, *cryptocurrency*, virtual account, e-money, QRIS, dan lain-lain dipakai oleh hampir seluruh unit usaha di seluruh negara Indonesia. Bahkan, di saat ini pembayaran untuk kendaraan umum pun dilakukan secara digital, MRT dan KRL menggunakan e-money, ojek *online* seperti Go-Jek menggunakan Gopay, dan lain-lain. Selain mudah digunakan, pengguna juga tertarik untuk menggunakan pembayaran digital karena banyak perusahaan *fintech* memberikan promo besar-besaran untuk pembayaran menggunakan metode pembayaran perusahaan tersebut.

Tetapi dengan seluruh kemudahan dan keuntungan pemakaian pembayaran digital ini, pembayaran digital memiliki beberapa kekurangan. Pembayaran digital tentunya memiliki *dependency* terhadap *server* yang bisa down kapan saja dan akan berdampak ke kecepatan dan tingkat kesuksesan pembayaran, seperti ketika melakukan pembayaran melalui kartu debit visa tentunya kesuksesan akan bergantung kepada bagaimana status *server payment gateway* tersebut, jika kebetulan di saat itu *server visa* sedang mengalami down, maka pembayaran akan sulit untuk dilakukan. Beberapa pembayaran digital juga bergantung ke koneksi internet *smartphone*, seperti dompet digital berbasis aplikasi *smartphone*, QRIS, virtual account melalui *m-banking*, dan lain-lain, ketika koneksi internet sulit untuk didapatkan maka akan sulit juga untuk melakukan pembayaran.

Kekurangan pembayaran digital yang kritical adalah soal keamanan, pembayaran digital memberikan banyak celah oknum kriminal untuk melakukan *fraud*. Seperti ketika sebuah aplikasi berbasis web yang menyimpan detail kartu kredit pengguna terjadi *breach*, maka ada risiko *hacker* untuk menggunakan detail kartu kredit ini untuk disalahgunakan

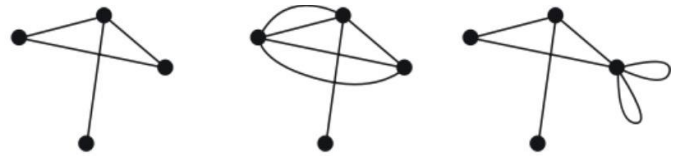
seperti dijual atau digunakan untuk pembelian barang-barang secara daring oleh *hacker* tersebut. Mungkin skema penyalahgunaan tersebut mudah ditanggulangi karena biasanya terdapat sistem notifikasi bank untuk setiap transaksi yang dibayar menggunakan kartu kredit tersebut lalu pembayaran bisa dibatalkan jika pemilik merasa transaksi tersebut tidak pernah dilakukan, tetapi akan sangat merepotkan jika barang sudah terlanjur terkirim karena pihak bank tentu saja tidak bisa begitu saja membatalkan pembayaran karena penjual sudah mengirimkan barang kepada *hacker*. Promo besar-besaran yang dilakukan perusahaan-perusahaan *fintech* juga memungkinkan untuk dicurangi, seperti misalnya sebuah marketplace memberikan *cashback* untuk pembelian barang, bisa saja pembelian dilakukan oleh kelompok *frauds* yang menjual dan membeli barang yang dijual sendiri untuk mendapatkan *cashback* secara cuma-cuma tanpa melakukan transaksi asli menggunakan platform *e-commerce* tersebut. *Frauds* juga sangat umum ditemukan dalam bisnis asuransi, salah satu contohnya adalah pemegang polis membentuk kelompok untuk melakukan rekayasa kecelakaan lalu lintas secara lengkap seperti penabrak, saksi, dan lain-lain lalu kelompok ini akan melakukan proses klaim asuransi dan mendapatkan uang gratis.

II. DASAR TEORI

A. Graf

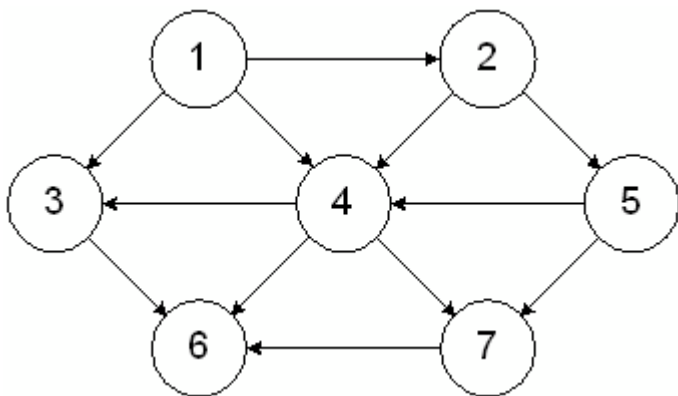
Graf, $G = (V, E)$, adalah objek yang terdiri dari dua elemen V dan E , dengan V adalah himpunan tidak kosong dari simpul dan E adalah himpunan dari sisi, penghubung antara 2 simpul.

Terdapat beberapa jenis graf, yang pertama adalah graf



Gambar 1. Graf (Sumber: WolframAlpha)

seederhana, yaitu graf yang tidak memiliki sisi yang menghubungkan 1 simpul dengan simpul itu sendiri, disebut dengan simpul gelang, dan tidak memiliki 2 sisi yang menghubungkan 2 simpul yang sama, disebut dengan simpul ganda. Kedua adalah graf tak-seederhana, graf yang mungkin memiliki simpul gelang dan simpul ganda dengan graf yang memiliki simpul gelang disebut sebagai graf semu dan graf yang memiliki simpul ganda disebut sebagai graf ganda.



Gambar 2. Graf Berarah (Sumber: faculty.cs.niu.edu)

Graf juga bisa dikelompokkan menjadi 2, yaitu graf tak-berarah, graf yang sisinya tidak memiliki orientasi arah, dan graf berarah, yaitu graf yang sisinya memiliki orientasi arah.

Terdapat beberapa terminologi umum dalam graf, yaitu sebagai berikut:

1. Ketetanggaan (Adjacency)
Dua buah simpul disebut bertetangga jika terdapat setidaknya 1 sisi yang menghubungkan 2 simpul tersebut.
2. Bersisian (Incidency)
Sembarang sisi $e = (S_1, S_2)$ disebut bersisian dengan simpul S_1 dan S_2 .
3. Simpul Terpencil (Isolated Vertex)
Sebuah sembarang simpul disebut terpencil (isolated) jika tidak ada setidaknya 1 sisi yang bersisian dengan simpul tersebut.
4. Graf Kosong (Null Graph)
Sebuah graf disebut graf kosong jika memiliki himpunan sisi yang kosong.
5. Derajat (Degree)
Derajat sebuah simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.
6. Lintasan (Path)
Lintasan graf adalah sebuah list tidak kosong berurut dari sisi berbeda yang menghubungkan 2 buah simpul.
7. Siklus (Cycle)
Siklus graf adalah lintasan yang dimulai dan berakhir di simpul yang sama.
8. Keterhubungan (Connection)
Dua buah simpul S_1 dan S_2 disebut terhubung jika terdapat lintasan yang menghubungkan simpul S_1 dan S_2 .
9. Upagraf (Subgraph)
Graf $G_1 = (V_1, E_1)$ disebut sebagai upagraf dari graf $G = (V, E)$ jika V_1 adalah himpunan bagian dari V dan E_1 adalah himpunan bagian dari E .
10. Upagraf Merentang (Spanning subgraph)
Graf $G_1 = (V_1, E_1)$ disebut sebagai upagraf merentang dari graf $G = (V, E)$ jika $V_1 = V$ dan E_1 adalah himpunan bagian dari E .
11. Cut set
Cut set dari sebuah graf adalah sebuah

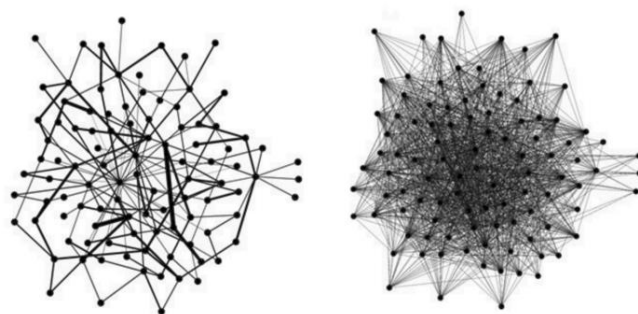
himpunan sisi yang jika sisi tersebut dihapus dari graf akan menghasilkan 2 upa-graf yang tidak terhubung.



Gambar 3. Graf Khusus (Sumber: spada.uns.ac.id)

Berdasarkan properti simpul dan sisi graf, graf memiliki bentuk beberapa bentuk khusus, sebagai berikut:

1. Graf Lengkap (Complete Graph)
Graf lengkap adalah bentuk graf khusus yang setiap simpulnya memiliki sisi ke setiap simpul lainnya kecuali simpul itu sendiri.
2. Graf Lingkaran
Graf lingkaran adalah bentuk graf khusus yang setiap simpulnya memiliki derajat 2.
3. Graf Teratur (Regular Graph)
Graf teratur adalah bentuk graf khusus yang setiap simpulnya memiliki derajat, d , yang sama.
4. Graf Bipartite (Bipartite Graph)
Graf bipartite adalah bentuk graf khusus yang simpul-simpulnya bisa dipisahkan menjadi 2 himpunan, V_1 dan V_2 , berdasarkan sisinya sehingga setiap sisinya akan menghubungkan 1 simpul di himpunan V_1 ke 1 simpul di himpunan V_2 .



Gambar 4. Sparse vs Dense Graf (Sumber: mathcenter.oxford.emory.edu)

Properti penting graf lainnya adalah densitas, yaitu seberapa banyak sisi dari sebuah graf. Graf dengan memiliki jumlah sisi yang jauh lebih sedikit dibandingkan jumlah sisi graf yang mungkin dimiliki disebut dengan graf sparse, sebaliknya graf yang memiliki jumlah sisi mendekati atau sama dengan jumlah sisi graf yang mungkin dimiliki disebut dengan graf dense. Properti densitas menjadi penting karena dalam analisis graf graf traversal, proses mengunjungi simpul-simpul dalam graf, adalah proses yang sangat sering dijumpai dan waktu yang diperlukan untuk melakukan traversal sangat dipengaruhi oleh densitas sebuah graf.

B. Basis Data

Basis data adalah sebuah himpunan dari informasi yang saling berhubungan. Awalnya basis data berbentuk tidak lebih dari kumpulan-kumpulan kertas berisi informasi yang dibutuhkan, tetapi dengan berkembangnya zaman basis data diubah menjadi sebuah sistem digital yang memungkinkan manusia untuk melihat data yang dibutuhkan dengan sangat cepat dan efisien. Basis data menjadi sangat penting karena pada dasarnya komputer adalah sebuah alat yang mengolah himpunan informasi untuk mendapat informasi/hal yang dibutuhkan.

Basis data memiliki banyak model yang digunakan sesuai kebutuhan sistem, beberapa yang paling populer adalah sebagai berikut:

1. Basis Data Relasional
2. Basis Data Berorientasi Dokumen
3. Basis Data Key-Value
4. Basis Data Graf

Pertama basis data relasional, basis data relasional memanfaatkan tabel sebagai representasi data. Dalam basis data relasional, 1 atau lebih kolom dipakai sebagai *unique identifier*, disebut sebagai *primary key* atau *compound key* jika tabel memiliki 2 atau lebih kolom yang menjadi kolom *primary key*, lalu kolom lainnya dipakai untuk menyimpan data yang tidak bersifat unik. Jika ingin menggambarkan hubungan antara 2 data di tabel yang berbeda umumnya digunakan *foreign key*, sebuah kolom dalam tabel yang berisi *primary key* dari tabel lain. Implementasi basis data relasional, seperti MySQL, PostgreSQL, SQLite, dan lain-lain, umumnya menggunakan sebuah bahasa kueri yang disebut sebagai Structured Query Language (SQL). Basis data relasional akan mengalami penurunan performa seiringnya penambahan jumlah dan hubungan antar data.

Kedua basis data berorientasi dokumen, basis data berorientasi dokumen menjadi sebuah dokumen menjadi unit penyimpanan data dengan beberapa format yang paling populer adalah Extensible Markup Language (XML), JavaScript Object Notation (JSON), Yet Another Markup Language (YAML), dan lain-lain. Data yang disimpan sebagai dokumen memudahkan proses mapping di dalam program, data juga bersifat fleksibel dalam artian dimungkinkan setiap data untuk memiliki field-field yang berbeda karena tidak disimpan dalam format yang kaku seperti tabel, lalu juga karena menggunakan format dokumen maka akan memudahkan proses replikasi basis data dalam proses *server horizontal-scaling* atau proses *disaster recovery*.

Ketiga basis data key-value, basis data ini memanfaatkan key sebagai acuan proses pencarian value dan seringkali memanfaatkan algoritma hashing sehingga proses pencarian akan sangat cepat. Tetapi karena tidak ada *fixed underlying* format data yang dipakai, basis data key-value umumnya tidak cocok untuk menyimpan data yang kompleks. Basis data key-value umumnya lebih cocok untuk dipakai sebagai sistem caching.

Keempat basis data graf, basis data ini memanfaatkan graf sebagai representasi data. Basis data ini sangat cocok untuk dipakai oleh sistem yang sangat bergantung kepada hubungan

antar data karena graf memiliki properti yang sangat terikat dengan hubungan antara simpul. Basis data graf umumnya menggunakan graf yang memiliki orientasi dan sisinya dianggap sebagai deskripsi bagaimana data, simpul, berhubungan dengan data lainnya.

Basis data graf memiliki keuntungan untuk beberapa kasus karena menggunakan graf sebagai representasi datanya. Misalnya dalam kasus aplikasi media sosial untuk karir dan profesional LinkedIn memiliki fitur *connection*, mirip seperti konsep teman, yang memungkinkan kita untuk melihat seberapa jauh kita terkoneksi dengan satu user, 1st-degree artinya kita memiliki koneksi langsung dengan user tersebut, 2nd-degree artinya ada koneksi 1st-degree kita yang memiliki koneksi langsung dengan koneksi tersebut, 3rd-degree artinya user tersebut memiliki koneksi dengan koneksi 2nd-degree kita. Lalu juga Facebook memiliki fitur friends atau friends of friends yang bekerja mirip seperti LinkedIn tetapi dengan abstraksi *connection* sebagai *friends*.

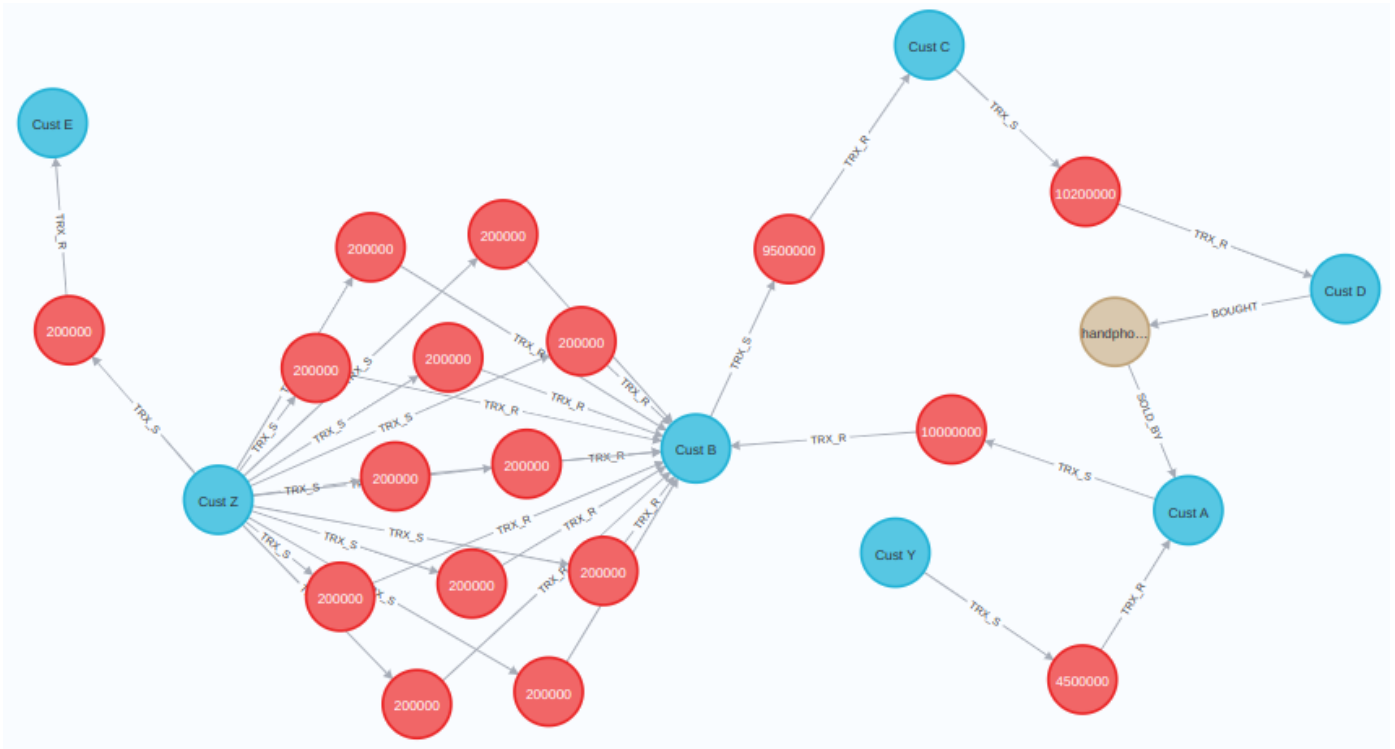
C. Neo4j

Neo4j adalah sebuah basis data graf *native* yang berarti dalam basis data ini setiap data memiliki informasi properti data dan hubungan antar data secara asli, bukan tabel dengan foreign key yang menghubungkan data dari 1 tabel ke tabel lainnya. Neo4j merepresentasikan setiap data menjadi sebuah simpul graf dan hubungan antar data, *relationship*, sebagai sisi yang memiliki orientasi. Neo4j memiliki sebuah bahasa kueri, seperti SQL, yang bernama Cypher.

Beberapa *basic syntax* dari bahasa kueri Cypher adalah sebagai berikut:

1. Match
Keyword match mencocokkan properti dari tiap simpul, relationship, label, pattern dari data, dan lain-lain.
2. Return
Keyword return mengartikan data apa yang ingin dikembalikan dari proses kueri yang dilakukan, bisa dalam bentuk simpul, relationship, pattern, dan lain-lain.
3. Where
Keyword where mengartikan syarat sebuah data termasuk kedalam himpunan data yang ingin dikembalikan.
4. Limit
Keyword limit artinya membatasi jumlah data yang ingin dikembalikan oleh keyword return.

Konsep yang sangat penting dalam menggunakan kueri Cypher adalah pattern. Menggunakan pattern artinya kita mencari data berdasarkan bentuk data.



Gambar 5. Gambar Data pada Basis Data Graf Neo4j

IV. APLIKASI GRAF

A. Fraud Kasus Pertama

Kasus pertama ini adalah sebuah *e-commerce* yang memiliki sistem kupon diskon untuk setiap pembelian yang memenuhi syarat, seperti misalnya diskon gratis ongkos kirim jika pertama kali menggunakan platform tersebut, diskon 5% untuk setiap pembelian diatas 2 juta rupiah, dan lain-lain. Di sisi lain, *e-commerce* ini juga memiliki sebuah sistem pembayaran yang dimiliki sendiri, sehingga bisa mengakses data riwayat transaksi setiap pengguna.

Dalam kasus ini, akan ada oknum yang memanfaatkan sistem kupon diskon 5% dengan cara menjual barang dengan harga yang sangat mahal lalu membuat pembelian semu agar mendapatkan kompensasi diskon dari platform *e-commerce*. Alur dari fraud ini adalah sebagai berikut (terurut sesuai nomor):

1. Penjual A yang menjual barang handphone dengan harga 10 juta rupiah
2. Ada pembeli B yang membeli barang handphone dari penjual A
3. Penjual A mengembalikan dana yang disamakan melalui beberapa transaksi dengan banyak akun di sistem pembayaran milik *e-commerce* tersebut. Akhirnya dana yang nilainya mirip dengan nilai transaksi pembelian semu tersebut akan kembali ke akun pembeli B.

Skenario ini memiliki beberapa syarat agar dianggap valid, yang pertama semua transaksi dilakukan dalam kurun waktu 3 hari, dan nilai setiap transaksi yang dilakukan

nilainya berada dalam range 20% nilai transaksi pembelian semua.

Agar bisa dilakukan analisis, dibuat basis data sampel untuk mencoba melakukan skenario diatas, dengan rincian desain basis data sebagai berikut:

Pertama, terdapat beberapa tipe simpul yang merepresentasikan tipe data, sebagai berikut:

1. Simpul Customer: simpul ini merepresentasikan customer, yaitu penjual, pembeli, dan seluruh pembuat transaksi.
2. Simpul Transaksi: simpul ini merepresentasikan transaksi yang dibuat di dalam sistem pembayaran *e-commerce* tersebut.
3. Simpul Produk: simpul ini merepresentasikan produk yang dijual oleh penjual.

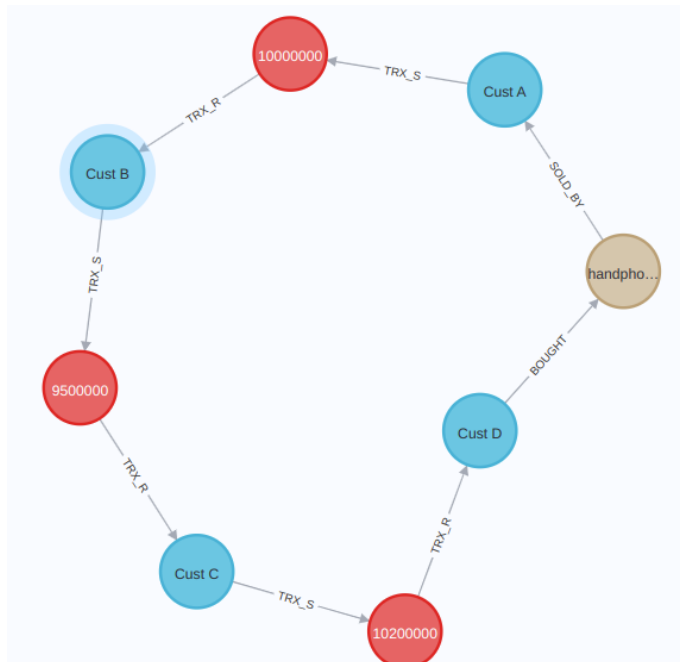
Kedua, terdapat beberapa tipe *relationship* atau sisi berarah yang merepresentasikan hubungan antara data, sebagai berikut:

1. Relationship TRX_R: relationship atau sisi ini mengartikan sebuah transaksi diterima oleh 1 customer.
2. Relationship TRX_S: relationship atau sisi ini mengartikan sebuah transaksi dikirim oleh 1 customer.
3. Relationship SOLD_BY: relationship atau sisi ini mengartikan sebuah produk dijual oleh 1 penjual.
4. Relationship BOUGHT: relationship atau sisi ini mengartikan sebuah produk dibeli oleh 1 pembeli.

Untuk mendapatkan data yang sesuai dengan syarat-syarat seperti yang sudah disebutkan kita akan melakukan *pattern-matching* menggunakan kueri Cypher seperti di gambar 6 dan didapatkan data seperti di gambar 7.

```
MATCH path = (seller_fraud)-[:SOLD]
->(product_fraud)-[:BOUGHT_BY]
->(buyer_fraud)-[r:TRX_R|TRX_S*..6]-(seller_fraud)
WHERE all(
  re in nodes(p)
  WHERE (
    re:Trx
    AND
    re.trxAmount >= (
      0.8 * product_fraud.price)
    AND
    re.trxAmount <= (
      1.2 * product_fraud.price)
    OR re:Customer or re:Product))
RETURN path
```

Gambar 6. Kueri Frauds Kasus Pertama



Gambar 7. Hasil Kueri Kasus Pertama

Sebenarnya dari gambar hasil tersebut sudah sangat terlihat bahwa terdapat sebuah keanehan bahwa terdapat cycle yang sangat kecil yang terdiri dari simpul produk, customer, dan transaksi dengan nilai setiap transaksi sangat mirip dengan nilai pembelian.

Bayangkan terdapat jutaan transaksi yang dilakukan setiap hari, maka akan sangat sulit untuk melihat pola yang terdapat pada gambar. Dengan menggunakan kueri seperti yang kita lakukan akan sangat memudahkan proses abstraksi pengawasan data karena yang terlihat hanya cycle yang mencurigakan untuk dilakukan pemeriksaan manual oleh pengawas *frauds*.

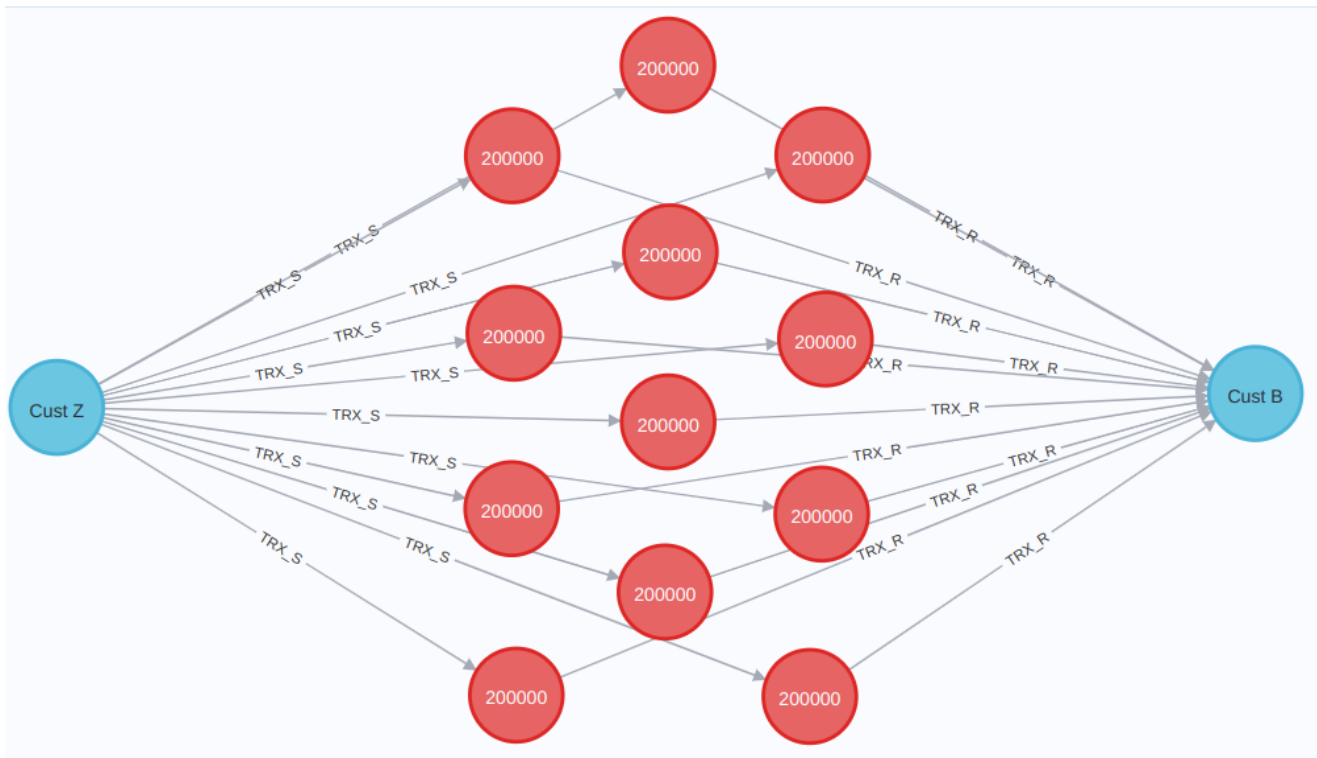
B. Fraud Kasus Kedua

Kasus kedua adalah sistem pembayaran yang sama dengan *e-commerce* kasus pertama. Kasus kedua ini adalah mencari pola mencurigakan dimana seorang customer melakukan transaksi yang sangat banyak ke customer yang sama dan dilakukan dalam kurun waktu 1 hari. Jumlah transaksi sebagai syarat aktivitas tersebut dianggap mencurigakan adalah 6 transaksi. Kasus ini sebenarnya tidak menandakan adanya *frauds*, tetapi kasus ini dimasukkan karena terdapat risiko transaksi ini dilakukan untuk tujuan yang tidak normal.

Untuk mendapatkan pola tersebut kita akan melakukan kueri *pattern-matching* seperti di dalam gambar 8. Lalu hasil kueri tersebut seperti di gambar 9.

```
MATCH path = (sender:Customer)-[:TRX_S]
->(trx)-[:TRX_R]
->(receiver:Customer)
WITH sender, receiver,
  count(path) as occ where occ > 6
MATCH path_p = (
  nx:Customer{name:sender.name})-[:TRX_S]
->(trx)-[:TRX_R]
->(ny:Customer{name:receiver.name})
RETURN path_p
```

Gambar 8. Kueri Frauds Kasus Kedua



Gambar 9. Hasil Kueri Frauds Kasus Kedua

V. CONCLUSION

Frauds sangat mudah terjadi di dalam sebuah sistem pembayaran digital dan bisa dilakukan dengan banyak cara. Salah satu aktivitas yang bisa dicurigai sebagai percobaan *frauds* adalah aktivitas transaksi yang sudah dibahas di bagian 4 paper ini. Dengan seluruh risiko yang ada maka sistem untuk menanggulangi atau mencegah *frauds* sangat dibutuhkan. Salah satu cara untuk menanggulangi atau mencegah *frauds* adalah dengan melakukan analisis dengan basis data graf seperti yang dilakukan di paper ini. Paper ini memiliki beberapa aspek yang masih bisa disempurnakan seperti melakukan *sum* atau jumlah transaksi yang dilakukan di kasus kedua sebagai syarat aktivitas bisa disebut mencurigakan, lalu juga kueri masih dilakukan secara manual seharusnya ada mekanisme otomatis eksekusi kueri yang mungkin di *trigger* melalui message broker untuk *real time execution* atau batch job ketika deteksi *frauds* tidak perlu dilakukan secara *real time*.

REFERENCES

- [1] Munir, Rinaldi. Bahan Kuliah IF2120 Matematika Diskrit Graf Bagian 1. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>. [Accessed Dec. 8, 2022]
- [2] Getting Started with Neo4j. Available: <https://neo4j.com/developer/get-started/>. [Accessed Dec.8, 2022]
- [3] Scifo, Estelle. Hands-On Graph Analytics with Neo4j.
- [4] Beaulieu, Alan. Learning SQL, 3rd Edition.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2022

Zidane

Zidane Firzatullah 13521163