

## Formalisasi *Graph* Multidimensi untuk Visualisasi Interaktif Proses Bisnis Terinspirasi OLAP

*Multidimensional Graph Formalization for OLAP-Inspired Business  
Process Interactive Visualization*

Alfa Ryano Yohannis, AlexWaworuntu, Tedi lesmana M.

Institut Teknologi dan Bisnis Kalbe, Jakarta

### Abstrak:

*Ide garis besar dari penelitian ini adalah bagaimana menggabungkan proses bisnis yang divisualisasikan sebagai graf dapat dieksplorasi secara interaktif oleh pengguna dengan melakukan operasi-operasi yang biasanya dapat dilakukan di OLAP browser. Kerangka kerja formal dibutuhkan sebagai panduan bagaimana seharusnya data proses bisnis disusun dan diproses sebelum divisualisasikan. Formalisasi graf, baik graf orde-pertama maupun graf orde tinggi, dijelaskan. Dengan menggunakan data eksperimen yang diambil dari data demo SAP Business One, visualisasi graf dari kedua orde ditampilkan untuk memperjelas hasil visualisasinya. Dengan adanya formalisasi graf multidimensi, visualisasi proses bisnis memiliki landasan formal yang cukup sebelum masuk ke dalam tahap pembangunan purwarupa.*

**Kata Kunci:** *formalisasi, kerangka kerja formal, graf multidimensi, visualisasi proses bisnis, SAP Business One*

### I. Pendahuluan

Perusahaan membutuhkan cara baru dan inovatif untuk memvisualisasikan proses bisnis mereka, visualisasi yang dapat memberikan fleksibilitas eksplorasi: menampilkan kinerja proses bisnis—baik struktur model maupun informasi tekstualnya, multiperspektif, mendukung personalisasi, dan juga mendukung abstraksi beragam tingkat.

Disisi lain, teknologi OLAP berkembang sebagai salah satu teknologi yang harus dimiliki dalam setiap paket komersial *business intelligence*. Interaktivitas yang dimiliki OLAP browser, dimana pengguna dapat melakukan operasi

*roll-up* (konsolidasi), *drill-down*, *slicing* dan *dicing*, menginspirasi penelitian ini untuk menghasilkan visualisasi proses bisnis, di mana pengguna dapat melakukan operasi-operasi tersebut secara interaktif.

Ide garis besar dari penelitian ini adalah bagaimana menggabungkan proses bisnis yang divisualisasikan sebagai graf dapat dieksplorasi secara interaktif oleh pengguna dengan melakukan operasi-operasi yang biasanya dapat dilakukan di OLAP browser. Makalah ini memaparkan kerangka kerja formal bagaimana seharusnya data proses bisnis disusun dan diproses sebelum divisualisasikan. Dengan demikian, visualisasi proses

bisnis memiliki landasan formal yang cukup sebelum masuk ke dalam tahap pembangunan purwarupa (*prototype*).

## II. Metode Penelitian

Penelitian ini dilakukan dengan metode sebagai berikut. Pertama-tama, studi pustaka dilakukan untuk mengetahui hasil-hasil penelitian yang berkaitan dengan formalisasi graf. Hasil-hasil penelitian tersebut dijadikan sebagai landasan teori untuk menghasilkan bentuk formal dari graf untuk visualisasi proses bisnis. Selanjutnya, data sampel pada SAP Business One dikaji apakah dapat dibentuk menjadi suatu struktur data graf yang merepresentasikan proses bisnis yang berjalan. Data SAP Business One kemudian dianalisis dengan diimplementasikan pada bentuk formal yang telah dibuat untuk mengetahui apakah data SAP Business One memenuhi kerangka formal. Jika memenuhi syarat kerangka formal, maka secara teoretis data SAP Business One dapat dijadikan sebagai studi kasus dan kerangka formal dapat dijadikan sebagai landasan dalam pembangunan kode program untuk visualisasi interaktif proses bisnis.

## III. Kajian Pustaka

Wattenberg dkk. [4] memperkenalkan PivotGraph suatu aplikasi visualisasi graf multidimensi di mana atribut-atribut dari entitas dapat dipetakan sebagai dimensi x dan y dan

membentuk simpul pada koordinat x dan y tersebut. PivotGraph juga mendukung operasi *roll-up* dan *selection* untuk eksplorasi graf. Keterbatasannya adalah PivotGraph tidak dapat menjawab pertanyaan mengenai graf yang memiliki tepi dengan beragam jenis dan atribut. Fleksibilitas interaksi pemilihan agregasi atributnya masih *combobox* dan ini masih dapat diganti dengan cara interaksi yang lebih baik, seperti *dragand drop* dan *doubleclickdrill-down/roll-up* seperti pada *OLAP browser*.

Zhao dkk. [1] mengajukan suatu model gudang data (*datawarehouse*) baru yang disebut GraphCube. Model graf yang mereka buat disebut juga sebagai jejaring multidimensi di mana dimensi-dimensi menjadi atribut dari simpul. Mereka mendefinisikan suatu konsep yang mereka sebut sebagai jejaring agregasi, yaitu *cuboid*—suatu *graphcube* yang berisi segala kemungkinan agregasi yang dapat dibentuk dari data jejaring yang tersedia. Untuk memperoleh graf, pengguna dapat melakukan 2 jenis operasi, yaitu *cuboidquery* dan *crossboidquery*. *Cuboidquery* untuk memperoleh jejaring dari *cuboid* tunggal dan *crossboidquery* untuk memperoleh *query* lintas *cuboid* atau bisa juga dianalogikan seperti operasi JOIN pada *cuboid*. Keterbatasannya adalah model mereka adalah model graf terbatas di mana tidak terdapat atribut pada tepinya dan mereka tidak menjelaskan arsitektur dan tahapan proses bagaimana

*cuboid* dihasilkan—hanya menjelaskan landasan formalnya saja.

Zhicheng dkk. [3] mengajukan suatu pendekatan umum untuk melakukan analisis visual berbasis jejaring multidimensi dan multitingkat pada data tabular multivariat. Mereka memperkenalkan aplikasi Ploceus, aplikasi yang memungkinkan pengguna membangun dan mentransformasi jejaring secara fleksibel, dan melakukan eksplorasi visual dan manipulasi jejaring secara dinamis. Keterbatasannya adalah mereka fokus kepada pendekatan umum, bukan kepada domain kinerja proses bisnis, dan fokus kepada graf berbentuk jejaring, bukan kepada graf berarah.

Rudolf dkk. [2] mengusulkan penggunaan SAPHANA sebagai teknologi basisdata NoSQL untuk menangani graf multidimensi, sehingga dapat menangani data graf dalam jumlah besar. Selain itu, dengan menggunakan *pattern-matching*, mereka mencoba memberikan kemudahan kepada pengguna melakukan operasi-operasi yang biasanya terdapat pada OLAP menggunakan *template*. Penelitian mereka masih dalam tahap pengerjaan pada saat makalah ini ditulis sehingga belum diketahui hasil akhirnya.

Keterbatasan dari kajian terkait saat ini, yaitu sebagian besar hanya fokus kepada graf tidak berarah atau jejaring, padahal untuk memodelkan proses bisnis yang dibutuhkan adalah graf berarah karena proses bisnis menyatakan

aliran kerja. Selain itu, sampai sejauh pengetahuan penulis, belum ada penelitian yang menggunakan graf multidimensi di domain kinerja proses bisnis dan tentang kepuasan pengguna terhadap penggunaan aplikasi graf multidimensi untuk analisis proses bisnis.

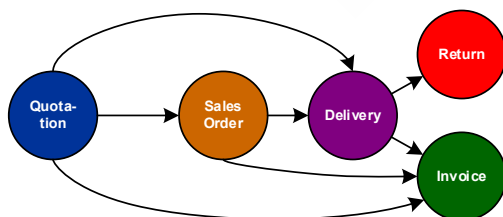
## IV. Hasil dan Pembahasan

### 4.1. Data Sampel dan Proses Penjualan Pada SBO

Data yang digunakan untuk menjelaskan formalisasi *graph* multidimensi terdapat pada Tabel 1 di bagian 8 Lampiran. Data tersebut merupakan cuplikan sebagian data transaksi penjualan pada basisdata demo *SBODemoAU* SBO dan merupakan hasil operasi JOIN antara tabel *header* dan tabel *detail* dari dokumen-dokumen proses pembelian. Kolom *HObjType* adalah kolom yang berisi tipe dokumen—*quotation, order, delivery, return*, atau *invoice*—proses penjualan yang dibuat. Kolom *HDocEntry* adalah kolom identitas dari dari setiap dokumen yang dibuat. Pada tingkat dokumen-dokumen proses penjualan, *HObjType* dan *HDocEntry* adalah identitas. Kolom *DLineNum* adalah kolom identitas baris yang ada dalam satu dokumen. Kolom *HDocDate* adalah tanggal kapan dokumen dibuat. Kolom *HCardName* adalah nama pelanggan yang tercantum pada dokumen penjualan. Kolom *DBaseType* adalah tipe dari

dokumen dasar yang menjadi acuan suatu dokumen dibuat. Kolom *DbaseEntry* dan kolom *DbaseLine* adalah identitas dokumen dan identitas baris dari dokumen dasar. Untuk kolom *DDescription*, *DQuantity*, *DPrice*, dan *DLineTotal* masing-masing adalah deskripsi produk yang dibeli, jumlah produk, harga produk, dan total nilai per baris yang harus dibayar oleh pelanggan. Kolom locale adalah *TupleId* yaitu kolom identitas yang bernilai unik untuk setiap tuple atau baris. localenantinya memudahkan pelacakan pada *graph* kompleks.

Proses penjualan pada SBO aktualnya terdiri dari proses *quotation*, *salesorder*, *delivery*, *invoicing*, *return*, *credit memo*, *downpaymentinvoice*, *downpaymentrequest*, dan *reserve invoice*, tetapi karena data dari keempat proses terakhir tersebut tidak pernah ada di basisdata demo SBO, maka proses-proses tersebut tidak diperhitungkan sebagai subproses dari proses penjualan. Dengan demikian, proses penjualan hanya terdiri dari proses *quotation*, *salesorder*, *delivery*, *invoicing*, dan *return*. Agar lebih jelas, proses penjualan dapat dilihat pada Gambar 1.



Gambar 1 Proses Penjualan pada SBO.

Proses biasanya dimulai dari

*quotation* di mana pelanggan masih menanyakan produk yang rencana akan dibelinya beserta dengan harganya. Jika proses *quotation* masuk masuk dalam kriteria pelanggan, kuotasibiasanya dilanjutkan dengan proses *salesorder*, yaitu proses di mana pelanggan sudah siap membeli produk dan terjadi kesepakatan beli-jual antara pelanggan dan perusahaan penjual. Produk yang akan dibeli selanjutnya disiapkan dan dibawa ke pelanggan melalui proses *delivery*. Setelah produk diantar, pembelian dilanjutkan dengan proses penagihan atau *invoicing*. Terkadang produk yang diantar tidak sesuai dengan pembelian, baik jenis ataupun jumlah, atau produk dalam keadaan rusak ketika diterima oleh pelanggan, sehingga produk harus dikembalikan ke perusahaan. Proses ini adalah proses *return*.

Untuk beberapa kasus, terkait dengan natur bisnis yang dijalankan, proses *order* ke *invoicing* tidak harus selalu melalui proses *delivery*, misalnya produk yang dibeli langsung oleh pelanggan di toko atau lokasi perusahaan tidak memerlukan proses *delivery*. Begitu juga dengan proses *quotation*, proses pembelian tidak harus selalu melalui proses *order*, *quotation* yang dibuat dapat langsung ke proses *delivery* atau *invoicing*, tergantung dari natur bisnis yang dijalankan.

Dengan menggunakan abstraksi proses penjualan Gambar 1, terlihat proses



*return* didahului oleh proses *delivery*. Dengan demikian, terdapat dokumen *return* yang dibuat berdasarkan dokumen *delivery*. Sebagai contoh, pada Tabel 1, dokumen *return* (baris 24) dengan  $(HObjType, HDocEntry, DLineNum) = ("Return", 12, 0)$  memiliki  $(DbaseType, DbaseEntry, DbaseLine) = ("Delivery", 220, 0)$  yang menyatakan bahwa dokumen *return* tersebut berasal dari dokumen *delivery* pada baris 17 yang memiliki  $(HObjType, HDocEntry, DLineNum) = ("Delivery", 220, 0)$ . Baris pada dua dokumen dengan tipe berbeda dapat dikatakan terhubung apabila  $t_1(DbaseType, DbaseEntry, DbaseLine) = t_2(HObjType, HDocEntry, DLineNum)$ , dengan  $t_1$  dan  $t_2$  menyatakan kedua baris tersebut. Dengan menggunakan penalaran yang sama, penelusuran proses, dasar dari, suatu dokumen dapat diketahui.

#### 4.2. GraphOrde-Pertama

Setiap baris yang terdapat pada Tabel 1 dapat dianggap sebagai fakta atau entitas (juga disebut sebagai baris (*row*) atau *record* atau *tuple*) terkecil di mana atribut-atribut yang dimilikinya direpresentasi sebagai kolom. Misalnya, pada setiap entitas pada masing-masing jenis dokumen penjualan, terdapat atribut *DDescription*, *DQuantity*, *DPrice*, dan *DlineTotal* yang menyatakan deskripsi produk yang dibeli, jumlah produk, harga produk, dan total nilai per baris yang harus dibayar oleh pelanggan. Setiap entitas

memiliki *key* atau identitas unik yang membedakannya dengan entitas lain. Pada Tabel 1, identitas unik tersebut adalah  $(HObjType, HDocEntry, DLineNum)$ .

Dalam konteks visualisasi proses bisnis, entitas satu akan dihubungkan dengan entitas lain sehingga terjalin hubungan satu sama lain membentuk *graph*. Kumpulan entitas-entitas dengan atribut yang sama membentuk suatu relasi berdasarkan kesamaan atribut-atribut tersebut. Sebagian besar definisi yang dibuat pada makalah berasal dari [3] dan diaplikasikan sesuai konteks penelitian ini.

**Definisi 1.** Relasi  $R$  adalah himpunan tuple  $T$ ,  $T = \{t_1, t_2, t_3, \dots, t_m\}$ , dengan himpunan dimensi  $D$ ,  $D = (d_1, d_2, d_3, \dots, d_n)$ . Setiap dimensi diasosiasikan dengan suatu tipe data, disebut juga domain dari  $d_i$  yang dinyatakan oleh  $dom(d_i)$ . Setiap tuple  $t$  merupakan daftar terurut dari nilai-nilai dimensi  $(v_1, v_2, v_3, \dots, v_n)$  di mana  $t.d_i = v_i$  dan  $v_i$  memiliki tipe data dari  $dom(d_i)$ . Sebagai contoh, pada Tabel 1, tuple  $T$  adalah setiap baris yang terdapat pada tabel,  $t_1 = ("Invoice", 192, 1, 7/4/2009, "SG Electronics", \dots, 243.75, 2,437.50)$ ,  $t_1.HcardName = "SG Electronics"$ ,  $D = (HObjType, HDocEntry, DLineNum, \dots, DPrice, DlineTotal)$ , dan  $dom(DlineTotal) = Double[3]$ .

**Definisi 2.** Suatu *Graph*  $G = (N, E)$  terdiri dari himpunan *node* atau simpul  $N$  dan himpunan *edge* atau tepian  $E$ . Setiap simpul  $n \in N$  merupakan pasangan dari  $(I, A)$

di mana  $l$  adalah string yang menyatakan nama dari simpul.  $A$  merupakan himpunan atribut yang mendeskripsikan simpul:  $A = (a_1, a_2, \dots, a_n)$ . Setiap atribut memiliki nilai  $n.a_i = v_i$ . Setiap tepian  $(n_1, n_2)$ , di mana  $e \in E = \{e_1, e_2, \dots, e_k\}$ , menghubungkan simpul satu dengan simpul lainnya, di mana  $n_1$  adalah simpul asal dan  $n_2$  adalah simpul tujuan. Tepian juga dapat diberi nilai  $w$  yang dinyatakan oleh  $w(n_1, n_2)$ [3].

**Definisi 3.** Suatu sub-himpunan dimensi  $\Delta$  merupakan himpunan bagian dari dimensi keseluruhan  $D$ . Subhimpunan dimensi dibutuhkan ketika membangun *graph* yang hanya memerlukan sebagian dari keseluruhan dimensi.  $\Delta = (d_1, d_2, \dots, d_i)$  di mana  $d_i \in D_s, d_i \in D.a$ . Jika terdapat himpunan dimensi  $\Delta$ , subtuplet  $t.\Delta$  dapat dibuat di mana  $t.\Delta = (v_1, v_2, \dots, v_i)$  yang merupakan proyeksi dari tuplet[3].

Pada Tabel 1, mendefinisikan  $\Delta = (HObjType, HdocEntry, DLineNum)$  maka  $t_2.\Delta = ("Invoice", 192, 2)$ .

#### 4.2.1 SimpulOrde-Pertama

**Definisi 4.** Simpulorde-pertama adalah simpul yang berasal dari baris tunggal pada tabel atau dengan kata lain label dan atributnya berasal dari satu tupel atau baris yang sama. Selain berupa nilai dari subdimensi, label atau atribut juga dapat berupa fungsi dari subdimensi[3].

$$n.l = t.\Delta \text{ atau } f(t.\Delta_i)$$

$$\square a_i \in n.A, n.a_i = t.\Delta_i \text{ atau } f(t.\Delta_i)$$

Sebagai contoh, berdasarkan Tabel

1, simpul dibuat dengan definisi berikut  $\square n \in N, n.l \in \{t.(HObjType)\} \ \& \ n.a_i \in \{n.(HObjType, HdocEntry, DlineNum, \dots, DDocTotal)\}$ . Karena Tabel 1 merupakan *self-referencingtable*, yaitu *foreignkey* tabel mengacu ke kolom/beberapa kolom *identity(ies)* pada tabel itu sendiri, contoh  $t_m(DbaseType, DbaseEntry, DbaseLine) = t_n(HObjType, HDocEntry, DLineNum)$ , maka diperlukan suatu himpunan simpul lagi dari tabel yang sama untuk membentuk simpul tujuan, dengan demikian tepian dapat dibentuk. Misalkan himpunan simpul tujuan adalah  $M$ , di mana  $\square m \in M, m.l \in \{t.(HObjType)\} \ \& \ m.a_i \in \{m.(HObjType, HdocEntry, DlineNum, \dots, DDocTotal)\}$ .

#### 4.2.2. TepianOrde-Pertama

Untuk membentuk tepianorde-pertama yang juga merupakan relasi antar-simpuls, maka  $N$  dan  $M$  perlu di-JOIN. Mengacu ke Tabel 1, maka  $N$  dan  $M$  dihubungkan melalui persamaan  $M.DbaseType = N.HobjType \ \& \ M.DBaseEntry = N.HDocEntry \ \& \ M.DbaseLine = N.DLineNum$ . Tepian yang terbentuk 29 buah (lihat Tabel 2) dengan basis adalah penggabungan antara  $n.locale$  dan  $m.locale$  di mana  $basis(m, n) = n.locale + m.locale$ . Basis dapat juga digunakan sebagai identitas tepian untuk mempermudah pelacakan.

#### 4.2.3. Contoh GraphOrde-Pertama

Sebagai contoh, Gambar 2 memperlihatkan graph-orde pertama.

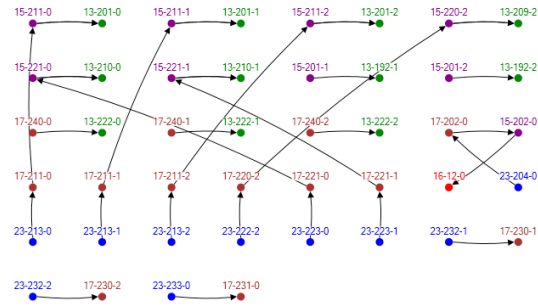
Data diambil dari Tabel 1 dan Tabel 2 di mana Tabel 1 sebagai simpul dan Tabel 2 sebagai tepian.

Untuk label yang ditampilkan, terdapat modifikasi dari penjelasan sebelumnya. Agar label yang ditampilkan tidak panjang, nilai string pada atribut *HObjType* dan *DObjType* diganti dengan angka sebagai berikut “Invoice” = 13, “Quotation” = 23, “Delivery” = 15, “Order” = 17 dan “Return” = 16. Label yang ditampilkan merupakan hasil dari penggabungan string antara atribut *HObjType*, *HdocEntry*, dan *DlineNum*, yaitu  $f(t, \Delta_i) = \text{concat}(t, \Delta_i) = \text{HObjType} + \text{"-"} + \text{HdocEntry} + \text{"-"} + \text{DlineNum}$  di mana  $t, \Delta_i = \{\text{HObjType}, \text{HdocEntry}, \text{DlineNum}\}$ , sesuai dengan Definisi 4.

Warna, salah satu dimensi spasial, simpul digunakan untuk merepresentasikan jenis dokumen penjualan, biru (*quotation*), coklat (*order*), ungu (*delivery*), hijau (*invoice*), dan merah (*return*). Penggunaan warna mempermudah identifikasi jenis dokumen pada simpul.

Terdapat banyak graph pada Gambar 2, di mana masing-masing graph mewakili satu proses penjualan per baris barang yang dijual. Pada gambar terdapat graph panjang (banyak simpul) dan graph pendek (sedikit simpul). Graph panjang menyatakan proses penjualan melibatkan banyak subproses penjualan sedangkan graph pendek sebaliknya. Dari Gambar 2 juga terlihat bahwa proses penjualan

tidak selalu dimulai dan diakhiri dari dan pada subproses tertentu saja, tetapi juga dapat dimulai dan diakhiri dari dan pada subproses beragam.



Gambar 2 Visualisasi graph orde-pertama proses penjualan pada SBO menggunakan aplikasi SimpulXL (data simpul dan tepian pada Tabel 1 dan Tabel 2).

### 4.3. GraphOrde Tinggi

Gambar 2 menampilkan graph pada tingkat abstraksi detail, sampai kepada tingkat perbaris barang pada dokumen penjualan. Tingkat abstraksi dapat dinaikkan, misalnya dinaikkan ke tingkat subproses penjualan sehingga mempermudah analisis pada tingkat proses bisnis.

Untuk mewujudkan abstraksi yang lebih tinggi, GraphOrde Tinggi diperlukan. GraphOrde Tinggi adalah graph yang dibangun lebih dari satu baris atau tupel pada tabel. Baris atau tupel penyusunnya perlu diagregasi sehingga simpul dan tepian tingkat lebih tinggi mencerminkan simpul dan tepian penyusunnya.

**Definisi 5.** Graph Orde Tinggi  $G' = (N', E')$  adalah *graph* yang bertransformasi dari *graph*  $G = (N, E)$  melalui proses agregasi simpul dan agregasi tepian [3].

**4.3.1. Agregasi Simpul**

**Definisi 6.** Himpunan simpul agregasi  $N' = (n_1', n_2', \dots, n_i')$  dengan simpul  $n_i'$  adalah agregasi dari himpunan simpul  $N_1 = \{n_{1,1}, n_{1,2}, \dots, n_{1,k}\}$ ,  $n_2'$  adalah agregasi dari himpunan simpul  $N_2 = \{n_{2,1}, n_{2,2}, \dots, n_{2,k}\}$ , dan  $n_i'$  adalah agregasi dari himpunan simpul  $N_i = \{n_{i,1}, n_{i,2}, n_{i,3}, \dots, n_{i,k}\}$ . Untuk mengubah himpunan simpul  $N_i$  menjadi simpul  $n_i'$  dibutuhkan suatu fungsi agregasi  $\mathcal{Z}_N(*)$  dengan  $n_i' = \mathcal{Z}_N(n_{i,1}, \dots, n_{i,k})$  di mana  $\{n_{i,1}, \dots, n_{i,k}\} \subseteq N$  dan  $n_i' \subseteq N'$  ( $\mathcal{Z}$  dibaca hori). Locale untuk simpul agregasi merupakan *union* dari locale simpul orde-pertama:  $locale(n_i') = locale(n_{i,1}) \cup \dots \cup locale(n_{i,k})$  [3].

**4.3.2. Agregasi Tepian**

Setelah agregasi simpul, tepian perlu dibentuk kembali, diturunkan berdasarkan tepian orde pertama dan simpul-simpul agregasi yang terbentuk.

**Definisi 7.** Himpunan tepian agregasi  $E' = (e_1', e_2', \dots, e_c')$  adalah himpunan di mana setiap tepiannya  $e_i' = e(n_a', n_b')$  menghubungkan simpul  $n_a'$ , merupakan agregasi dari himpunan simpul  $N_a = \{n_{a,1}, n_{a,2}, \dots, n_{a,k}\}$ , ke simpul  $n_b'$ , merupakan agregasi dari himpunan simpul  $N_b = \{n_{b,1}, n_{b,2}, \dots, n_{b,k}\}$ . Setiap anggota  $E'$ , yaitu  $e_i'$ , diperoleh dengan fungsi agregasi  $\mathcal{Z}_E(n_a', n_b')$  yang mengagregasi semua tepian yang menghubungkan anggota  $N_a$  ke anggota  $N_b$ , di mana  $e_i' = \mathcal{Z}_E(n_a', n_b') = \cup \{ \mathcal{Z}_E(n_a, n_b) \mid n_a \subseteq N_a, n_b \subseteq N_b \}$ . Jika  $\mathcal{Z}_E(n_a', n_b') = \emptyset$ ,  $n_a'$  tidak terhubung ke  $n_b'$  [3].

$\cup \{ \mathcal{Z}_E(n_a, n_b) \mid n_a \subseteq N_a, n_b \subseteq N_b \}$ . Jika  $\mathcal{Z}_E(n_a', n_b') = \emptyset$ ,  $n_a'$  tidak terhubung ke  $n_b'$  [3].

**4.3.3. Pengirisan dan Pengotakan**

Pengirisan dan pengotakan (*Slicing and Dicing*) adalah proses pengkodisian atau pemfilteran baris atau tupel. Secara formal, untuk membuat suatu himpunan simpul  $N$  dari himpunan dimensi  $\Delta_n$  menggunakan himpunan dimensi pengirisan dan pengotakan  $\Delta_s$  adalah sebagai berikut:

$$\{ n \subseteq N, n.l \subseteq \{ t.\Delta_n \mid \text{COND}(t.\Delta_s) \} \}$$

Persamaan tersebut memfilter tabel asal  $\Delta_n$  berdasarkan kesamaan dimensi yang juga terdapat pada  $\Delta_s$ , dengan nilai kondisi yang didefinisikan oleh pengguna. Pemfilteran dan pengotakan membentuk suatu sub-himpunan simpul baru [3]. Tepian juga akan dibuat ulang berdasarkan sub-himpunan simpul baru tersebut.

**4.3.4. Pembobotan Tepian**

Pilihan awal (*default*) bobot tepian adalah kardinalitas dari tepian tersebut  $w(n_1, n_2) = |e(n_1, n_2)|$  atau frekuensi kemunculan dari tepian atau sama dengan fungsi COUNT yang terdapat di DBMS (*Database Management System*).

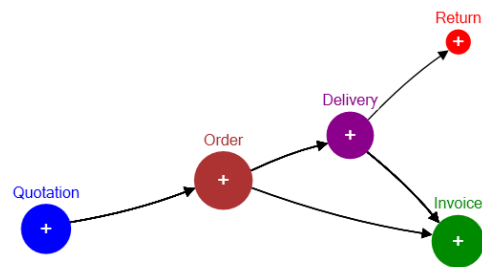
Bobot dari tepian juga dapat ditentukan menggunakan fungsi  $\mathcal{H}$  (baca: khei), yaitu  $w(n_1, n_2) = \mathcal{H}(e.\Delta)$  dengan menganggap  $e$  juga memiliki himpunan dimensi yang dapat digunakan sebagai input fungsi  $\mathcal{H}$  [3]. Fungsi  $\mathcal{H}$



dapat melakukan operasi-operasi seperti COUNT, SUM, AVERAGE, MAXIMUM, MINIMUM, atau fungsi lainnya.

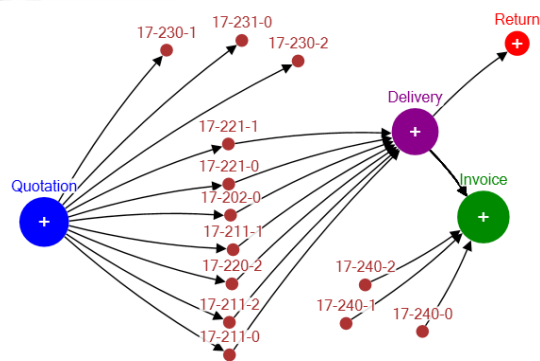
#### 4.3.5. Contoh Graf Orde Tinggi

Graf orde pertama yang ditampilkan pada Gambar 2 dapat ditransformasi menjadi graf orde tinggi. Misalnya, jika tiap simpul, representasi baris pada tiap dokumen subproses penjualan, diagregasikan berdasarkan jenis dokumen prosesnya, maka graf yang dihasilkan dapat dilihat pada Gambar 3. Total simpul yang dihasilkan hanya lima, yaitu simpul dengan label *quotation*, *order*, *delivery*, *invoice*, dan *return*, sesuai dengan jumlah jenis dokumen subproses penjualan. Pada Gambar 3, dimensi spasial juga dapat dimanfaatkan untuk merepresentasikan bobot simpul di mana ukuran diameter simpul orde tinggi mengikuti jumlah simpul orde pertama yang diwakilinya. Makin banyak simpul orde pertama yang diwakilinya, makin besar ukuran diameter simpul tersebut dan sebaliknya. Pada Gambar 3, simpul terbesar adalah simpul *order*. Jika menilik pada Tabel 2 dan Gambar 2, maka jumlah simpul yang memiliki tepian, dikelompokkan berdasarkan jenis dokumennya, adalah sebagai berikut:  $|N_{quotation}| = 10$ ,  $|N_{order}| = 13$ ,  $|N_{delivery}| = 9$ ,  $|N_{invoice}| = 11$ ,  $|N_{return}| = 1$ .  $|N_{order}|$  memiliki nilai paling besar 7, sesuai dengan Gambar 3, simpul *order* berdiameter paling besar.



orde pertama diagregasi berdasarkan jenis dokumennya.

Jika tidak semua simpul pada graf diagregasi, misalkan semua simpul diagregasi kecuali simpul dengan jenis dokumen *salesorder*, maka graf yang dihasilkan terdapat pada Gambar 4. Mode visualisasi seperti ini memberikan fleksibilitas ke pengguna untuk melakukan analisis drill-down terisolasi, melihat lebih detail ke salah satu proses sementara proses (simpul) lain tetap berada pada abstraksi yang lebih tinggi.



Gambar 4 Graf orde tinggi di mana simpul dengan jenis dokumen *salesorder* tidak diagregasi.

Pada Gambar 4, simpul-simpul berjenis dokumen *order* dapat dikelompokkan menjadi tiga. Pertama, simpul-simpul yang berasal dari proses *quotation* tetapi belum diproses lebih

lanjut menjadi dokumen subproses penjualan lainnya. Ini menunjukkan masih terdapat barang pada dokumen *salesorder* yang belum diproses. Kedua, simpul-simpul yang berasal dari simpul *quotation* dan sudah diproses menjadi dokumen penjualan lainnya. Ini menunjukkan barang pada dokumen *salesorder* yang sudah diproses menjadi *delivery*. Ketiga, simpul-simpul yang tidak berasal dari simpul *quotation* dan langsung diproses menjadi *invoice*, tanpa proses *delivery*. Ini kemungkinan menunjukkan pelanggan mengambil barang langsung di tempat penjualan, tanpa perlu proses *delivery*.

## V. Kesimpulan

Melalui makalah ini, kerangka kerja formal graf multidimensi untuk keperluan visualisasi proses bisnis interaktif yang terinspirasi oleh OLAP telah dipaparkan. Baik graf orde-pertama dan graf orde tinggi telah dijelaskan, termasuk juga bagaimana proses agregasi dan pembobotan simpul dan tepian beserta pengirisan dan pengotakan (*slicing and dicing*). Dengan menggunakan data yang terdapat pada basis data demo SAP Business One, graf orde-pertama dan graf orde tinggi divisualisasi sehingga bentuk dari kedua graf dapat terlihat lebih jelas, termasuk pembobotan simpul dan tepian beserta pengirisan dan pengotakan juga dimasukkan pada visualisasi tersebut. Kerangka kerja formal memberikan

panduan bagaimana seharusnya data proses bisnis disusun dan diproses sebelum divisualisasikan. Dengan demikian, visualisasi proses bisnis memiliki dasar formal yang cukup sebelum masuk ke dalam tahap pembangunan purwarupa.

Kerja selanjutnya dari penelitian ini adalah bagaimana membangun struktur data yang merepresentasi graf, termasuk simpul dan tepiannya, dan bagaimana struktur data tersebut mengakomodasi operasi roll-up, drill-down, slicing and dicing. Setelah itu, perancangan bagaimana arsitektur sistem dan aplikasi secara keseluruhan dan dilanjutkan dengan pembangunan purwarupa.

## Referensi

- [1] P. Zhao, X. Li, D. Xin and J. Han, "Graph Cube: On Warehousing and OLAP Multidimensional Networks," in *SIGMOD International Conference on Management of Data*, 2011
- [2] M. Rudolf, M. Paradies, C. Bornhövd and W. Lehner, "SynopSys: Large Graph Analytics in the SAP HANA Database Through Summarization," 2013.
- [3] L. Zhicheng, S. B. Navathe and J. T. Stasko, "Network-based visual analysis of tabular data," in *In Visual Analytics Science and Technology (VAST)*, 2011.

- [4] M. Wattenberg, “Visual Exploration of Multivariate Graphs,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006.

### **Pernyataan**

Makalah ini merupakan bagian dari penelitian unggulan jurusan berjudul “Visualisasi Proses Bisnis untuk Pemantauan Kinerja Proses Bisnis: Studi Kasus SAP Business One (SBO)” yang didanai oleh Kalbis Institute dengan nomor P3KM/ITBK/XI/2013.