

Implementasi Algoritma Konstruksi Relasi Objek Transitive Closure Berdasarkan Adjacency Hyperedges Matrix

Soetrisno¹⁾, Winadi Wiratama²⁾, Frans Panduwinata³⁾

¹⁾Fakultas Ilmu Komputer
Universitas Pelita Harapan

Karawaci, Tangerang, 021-5460901

e-mail: ¹⁾sutrisno.fik@uph.edu, ²⁾winadiw@gmail.com, ³⁾frans.panduwinata.fik@uph.edu

Abstrak

Relasi antar objek dapat dihubungkan melalui satu fitur atau atribut tunggal secara individual. Namun, dalam makalah ini relasi objek berdasarkan himpunan atribut objek yang dimiliki oleh objek yang bersangkutan. Relasi objek yang ditinjau adalah suatu relasi objek yang mencakup (covers) objek lainnya dari suatu koleksi objek yang diamati berdasarkan himpunan atribut yang dimilikinya. Oleh karenanya, relasi objek-objek tersebut membentuk suatu graph koneksi objek yang bersarang (nested) dikenal sebagai *partially order set* atau *lattice*. Suatu objek dapat memiliki koneksi langsung (transitive) dan path ke objek lainnya. Konstruksi koneksi langsung semua objek merupakan suatu transitive closure. Algoritma konstruksi transitive closure yang dikembangkan penulis dalam makalah ini berdasarkan model hypergraph yaitu Adjacency Hyperedges Matrix. Dalam koleksi objek yang dievaluasi jika ada relasi objek dapat merupakan suatu graph terkoneksi, atau juga membentuk lebih dari satu forest graph.

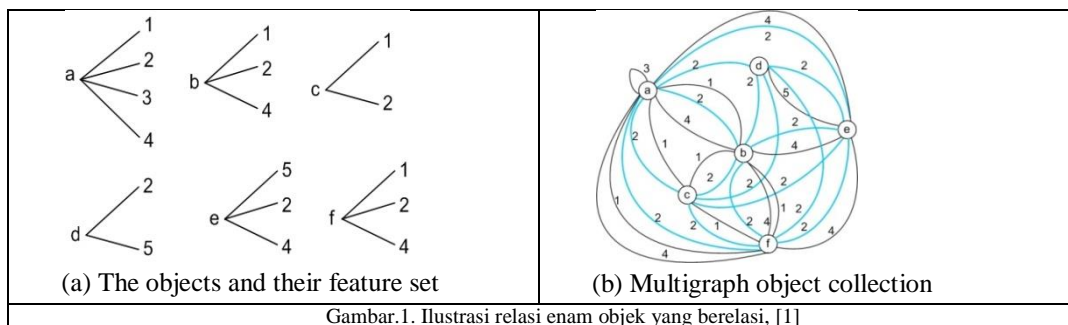
Kata kunci: hypergraph, data mining, identification, adjacency hyperedges, clustering

1. Pendahuluan

Relasi objek dapat dihubungkan melalui satu fitur atau atribut tunggal secara individual. Jika semua objek yang terhimpun dalam sebuah koleksi objek memiliki relasi pada satu atribut tunggal tersebut, maka membentuk suatu *complete graph* (*clique*). Sebuah objek yang dideskripsikan oleh satu himpunan atribut kemungkinan memiliki relasi dengan objek lainnya pada satu atau lebih atribut deskripsi objek. Relasi objek tersebut akan membentuk sebuah representasi *multigraph* [1].

Sebuah contoh relasi objek yang membentuk sebuah multigraph yaitu relasi objek bentuk geometri dua dimensi dan tiga dimensi. Objek-objek tersebut yaitu lingkaran, elips, dan persegi panjang, ketiganya berbeda, namun memiliki juga sejumlah kesamaan. Demikian pula pada objek geometri tiga dimensi yaitu bola dengan objek abstrak tiga dimensi. Beberapa kesamaan di antara mereka adalah memiliki luas, dan keliling, pada geometri dua dimensi, dan luas serta volume pada geometri tiga dimensi.

Gambar 1, menunjukkan multigraph relasi objek geometri di atas. Elipse dinotasikan sebagai a, lingkaran dinotasikan oleh b dan f, objek abstrak dua dimensi oleh c, objek abstrak tiga dimensi oleh d, dan geometri bola dinotasikan oleh e. Atribut yang menjadi ciri atau deskripsi objek yaitu keliling dinotasikan dengan angka 1, luas dinotasikan dengan angka 2, delta radius oleh angka 3, radius oleh angka 4, dan volume dinotasikan oleh angka 5. Gambar 1(b), merupakan *multigraph* dari koleksi objek a, b, c, d, e, dan f.



Gambar.1. Ilustrasi relasi enam objek yang berelasi, [1]

Enam objek tersebut terhubung atau memiliki relasi pada atribut luas secara individual, yang dinotasikan oleh angka 2. Enam objek tersebut belum tentu memiliki relasi jika ditinjau berdasarkan satu himpunan atribut yang menjadi ciri objek yang bersangkutan. Antara objek a dengan objek e, keduanya hanya terhubung oleh atribut 2 (luas) dan atribut radius (4), jadi hanya merupakan *partially similar objects* [1][2][3]. Dengan demikian, pada koleksi enam objek tersebut, objek-objek mana sajakah yang membentuk relasi berdasarkan himpunan deskripsi objeknya.

Pada proses identifikasi umumnya dilakukan dengan membandingkan himpunan atribut yang dimiliki oleh suatu objek yang diobservasi terhadap himpunan kelompok atribut pada suatu kelompok objek referensi [4]. Dan, pada proses identifikasi seperti *classification* menggunakan *dataset* domain sebagai referensi, sama dengan objek yang akan dievaluasi. Pada proses *clustering* yaitu melakukan pengelompokan objek-objek berdasarkan kesamaan atribut yang ada dalam *dataset*. Ada kemungkinan satu atau beberapa atribut dari sebuah objek, sama dengan objek lainnya, dan sebagian atribut lainnya sama dengan objek lain lagi, atau ada kemungkinan himpunan atribut objek tersebut dicakup (*subset*), atau mencakup (*superset*) himpunan atribut objek lainnya. Himpunan deskripsi objek a mencakup (*superset*) himpunan deskripsi objek b (lingkaran sebagai *subset*).

Pada *multigraph* di atas (Gambar 1), objek direpresentasikan oleh vertex, dan atribut direpresentasikan oleh edge. Pada paper [1][2][3], penulis mengembangkan model representasi koneksi objek berupa *hypergraph* yang dinamakan *Adjacency Hyperedges Matrix*. *Hypergraph* dipergunakan pada sejumlah domain persoalan, seperti *pattern recognition* oleh Ren dalam disertasi PhD-nya, penggunaan *hypergraph* pada *classification* oleh Sun et al., Saliu dan Ruhe menggunakan *hypergraph* dengan *hyperedges*-nya pada software engineering [5][6][7][8]. Pada paper [1], penulis mengembangkan model untuk identifikasi relasi objek komposisi yang transitive closure. Paper ini merupakan kelanjutan dari [1], model tersebut diimplementasi yang diujikan khususnya pada *dummy data* dengan jumlah objek yang lebih banyak.

Hypergraph yaitu *Adjacency Hyperedges Matrix* digunakan untuk menunjukkan bahwa pada sebuah dataset yaitu koleksi objek yang terepresentasi oleh himpunan atribut sebagai deskripsi objeknya merupakan *partially similar*. Relasi *inclusion* ini khususnya *proper superset/subset* adalah relasi koneksi objek yang *partially similar*. Dalam paper ini, relasi tersebut dinamakan relasi objek komposisi, yaitu suatu relasi berdasarkan himpunan deskripsi objek yang memenuhi operator relasi *superset*. Relasi objek antara objek a dan b merupakan suatu relasi *partially order set (poset)*. Salah satu karakteristik suatu poset yaitu *transitive edge*, suatu koneksi langsung *vertex* (objek) ke objek lainnya, implikasi dari adanya *path* objek ke objek yang *transitive* tersebut. Bilamana semua objek dalam koleksi objek memiliki *transitive edge* secara eksplisit sepanjang memenuhi operator relasi *poset (superset)*, maka *poset* tersebut merupakan suatu *transitive closure*. Penulis mengembangkan algoritma untuk mengidentifikasi dan mengkonstruksi relasi objek komposisi yang transitive closure. Pada paper ini algoritma tersebut diimplementasikan pada beberapa dataset.

2. Metode Penelitian

2.1. Adjacency Hyperedges Matrix

Hypergraph merupakan bentuk umum (*generalization*) *graph*. Suatu *hypergraph* $H = (V, E)$, V adalah himpunan vertices $v \in V$, dan E adalah himpunan *hyperedges* $e \in E$. *Hyperedge* e merupakan sub-himpunan nir-kosong dari V , yaitu $\cup_{e \in E} = V$ [8]. *Hypergraph* H dapat direpresentasikan sebagai *incident matrix* dengan ordo $|V| \times |E|$. Berikut, Tabel 1, adalah *hypergraph* $H = (V, E)$ pada Gambar 1. Gambar 1, merupakan *multigraph* koleksi objek a, b, c, d, e, dan f. *Multigraph* tersebut dapat direpresentasikan dalam bentuk *hypergraph* [1].

Tabel 1. Matriks incident pada *hypergraph* H

	1	2	3	4	5
a	1	1	1	1	
b	1	1		1	
c	1	1			
d		1			1
e		1		1	1
f	1	1		1	

Tabel 2. Matriks *Hypergraph Laplacian* pada Tabel 1

	1	2	3	4	5
1	4	4	1	3	
2	3	6	1	4	2
3	1	1	1	1	
4	2	3	1	4	1
5		2			2

Sebuah atribut yang direpresentasikan oleh *hyperedge* yaitu kolom pada Tabel 1. Tabel 1 merupakan representasi *hypergraph* H berupa *incident matrix*. Sebuah vertex memiliki *incident* ke *hyperedge* e dinotasikan sebagai $h(v, e) = 1$, dan jika tidak memiliki *incident*, dinotasikan $h(v, e) = 0$. Himpunan vertices yaitu objek $V = \{a, b, c, d, e, f\}$, dan himpunan *hyperedges* $E = \{1, 2, 3, 4, 5, 6\}$.

Incident $h(a, 4) = 1$, artinya *vertex* atau objek a memiliki atribut (*hyperedge*) yang dinotasikan dengan angka 4, tetapi *vertex* a tidak memiliki *incident* dengan *hyperedge* 5, $h(a, 5) = 0$. Model *hypergraph* yang dikembangkan merupakan *hypergraph* berbobot, dan untuk tiap *hyperedge* yang memiliki *incident* $h(v, e) = 1$, diberi bobot 1, lihat Persamaan (1).

$$w(e) = \begin{cases} 1 & \exists v \in V, h(v, e) = 1 \\ 0 & \text{lainnya} \end{cases} \quad (1)$$

Sebuah *hypergraph* memiliki dua jenis kardinalitas (*degree*) yaitu *vertex degree* dan *hyperedge degree* [1][2][3]. *Vertex degree* v adalah jumlah bobot *hyperedge* yang ada pada *vertex* v , dinotasikan sebagai $d(v)$, dan formulasinya dapat dilihat pada [1][2][3]. Kardinalitas *vertex* (*vertex degree*) v merupakan kardinalitas baris *incident matrix* pada *hypergraph* H , contoh lihat Tabel 1. Kardinalitas *hyperedge* (*hyperedge degree*) adalah banyaknya *vertex* yang menjadi anggota *hyperedge* e , dan dinotasikan $\delta(e)$, yaitu banyaknya elemen pada kolom *incident matrix*, lihat contoh Tabel 1.

Dari matriks suatu *hypergraph* dapat dibentuk matriks *adjacency* yang dinamakan *Laplacian hypergraph*. *Laplacian hypergraph* adalah sebuah matriks terdiri dari matriks *adjacency* A , dan matriks *degree* D , dan *Laplacian hypergraph* secara umum dinotasikan sebagai $L = D + A$. *Adjacency hyperedges matrix* merupakan model *hypergraph* yang dikembangkan penulis [3]. *Adjacency hyperedges matrix* dinotasikan sebagai A_e , merupakan suatu *adjacency matrix* pada *Laplacian hypergraph*. *Laplacian hypergraph* dinotasikan sebagai $L = A_e + D_e$. Formulasi model *Laplacian hypergraph* yang dikembangkan dapat dilihat pada Persamaan (2). Matriks *Laplacian* L merupakan perkalian matriks transpose *hypergraph* H yaitu H^T , *hypergraph* H , dan matriks bobot *hyperedge* yaitu matriks identitas. Matriks D_e merupakan matriks diagonal pada L , yang elemennya merupakan *hyperedge degree* $\delta(e)$.

$$L = \{l_{i,j}\} = H^T H W = A_e + D_e \quad (2)$$

Formulasi *adjacency hyperedges matrix* A_e yang mana untuk setiap *incident* $h(v, e) = 1$ diberikan bobot 1, maka elemen matriks *Laplacian hypergraph* dapat pula diformulasikan berupa kardinalitas *intersection set* antar dua *hyperedges* e_i dan e_j . Formulasinya dapat dilihat pada Persamaan (3).

$$L = \{l_{i,j}\} = A_e + D_e = |e_i \cap e_j|, \text{ dan } A_e \text{ jika } i \neq j, \text{ dan } D_e \text{ jika } i = j \quad (3)$$

Tabel 2 merupakan *Laplacian hypergraph* H pada Tabel 1, yaitu $H^T H W$, terdiri dari matriks diagonal D_e , dan *Adjacency hyperedges matrix* $A_e = H^T H W - D_e$. Elemen $e_{i,i}$ matriks diagonal D_e adalah *hyperedge degree* e_i , yaitu $\delta(e_i)$. Pada Tabel 2 diperlihatkan elemen $e_{2,2} = 6$, merupakan kardinalitas *hyperedge* fitur (atribut) 2, lihat *hypergraph* H pada Tabel 1. Elemen $e_{i,j}$ matriks *adjacency* A_e adalah *adjacency* antar dua *hyperedges* e_i dan e_j . Nilai elemen $e_{i,j}$ tidak nol jika keduanya memiliki sedikitnya satu *vertex* (*non-empty subset*), $e_i \cap e_j \neq \emptyset$. Elemen $e_{2,4} = 4$, merupakan kardinalitas *intersection set* antar atribut 2 dan atribut 4, yang anggotanya adalah himpunan *vertices* $\{a, b, e, f\}$, lihat Tabel 1.

2.2. Koneksi Objek Partially Similar

Berdasarkan formulasi Persamaan (2) dan (3) mengenai *Laplacian hypergraph*, maka dikembangkan sejumlah formulasi untuk koneksi objek yang ada pada *hypergraph* atau *subhypergraph*, yaitu identik, berbeda, dan memiliki kesamaan sebagian (*partially similar objects*). Sebuah *subhypergraph* $K \subseteq S$ dari *hypergraph* $H = (S, F)$ adalah *hypergraph*. S adalah himpunan *vertices*, dan F adalah himpunan *hyperedges* yang mana setiap *hyperedge* merupakan *non-empty subset* dari S . Formulasi dari jenis-jenis koneksi objek dalam koleksi objek S dapat dilihat pada [1][2][3].

Hypergraph $H(K) = (K, F(K))$ adalah *subhypergraph* dari *hypergraph* $H = (S, F)$ untuk $K \subseteq S$. $F(K)$ merupakan adalah himpunan *hyperedges* yang memiliki sedikitnya satu K sebagai anggota *hyperedge* yang bersangkutan, dan *hyperedge* lainnya yang tidak memiliki K sebagai komplemen. *Hypergraph* $H_K = (K, F_K)$, yang mana himpunan *vertices*-nya adalah K , dan F_K merupakan himpunan *hyperedges* hanya memiliki K sebagai anggotanya, $\forall e_i \in F_K, e_i = K \cap F_i(K), 1 \leq |e_i| \leq |K|$. Kardinalitas $F(K)$ dan F_K sama, yaitu $|F_K| = |F(K)|$.

Relasi antar objek pada *dataset* K yang mana objek-objek dalam K merupakan relasi yang memiliki sebagian kesamaan (*partially similar*) ditunjukkan pada Persamaan (4), dan Persamaan (5). Sumasi atribut objek-objek di K yang dinotasikan $n(K)$ selalu lebih besar dari kardinalitas himpunan *hyperedges* F_K , lihat Persamaan (4). Ada sedikitnya satu *hyperedge* di F_K memiliki kardinalitas lebih dari 1, artinya ada suatu *hyperedge* menghubungkan dua atau lebih objek (*vertices*) di K , lihat Persamaan (5).

Adapun $n(K)$ adalah jumlah (sumasi) atribut pada himpunan objek K merupakan sumasi *vertex degree* objek di K.

Berdasarkan Persamaan (4) dan Persamaan (5), maka koleksi objek $K \subseteq S = \{a, b, c, d, e, f\}$ merupakan koneksi objek yang *partially similar*. Algoritma untuk menetapkan suatu koleksi objek merupakan identik, partially similar, ataupun berbeda dapat dilihat pada [1][2][3]. Penulis berfokus pada koneksi objek yang *partially similar*, karena pada koneksi objek ini ada kemungkinan memiliki relasi *inclusion* pada koleksi objek tersebut.

$$n(K) > |F_K| \quad (4)$$

$$\exists e_i \in F_K, \delta(e_i) > 1 \quad (5)$$

Dari koleksi objek yang dievaluasi, maka koneksi atau relasi objek dapat ditinjau berdasarkan <atribut, nilai-atribut>, juga dapat ditinjau hanya berdasarkan atribut saja. Objek yang semula tidak memiliki relasi objek komposisi, mungkin membentuk relasi objek komposisi berdasarkan atribut saja.

2.3. Relasi Objek Komposisi yang Transitive Closure

Relasi objek yang memiliki sebagian kesamaan (*partially similar*) memenuhi Persamaan (4) dan Persamaan (5). Sedikitnya ada satu *hyperedge* $e_i \in F_K$ pada *hypergraph* $H_K = (K, F_K)$, memiliki kardinalitas lebih dari 1. Implikasinya adalah kardinalitas himpunan F_K lebih kecil dari sumasi atribut pada K, yaitu $n(K)$. Jika himpunan K hanya memiliki dua objek, dan relasi di antara keduanya adalah *partially similar*, maka ada kemungkinan dua objek tersebut membentuk suatu relasi *inclusion* yaitu *proper subset/superset*. Relasi *inclusion* terbentuk hanya jika kardinalitas (*vertex degree*, $d(v)$) dua objek tersebut berbeda.

Artinya ada sedikitnya satu atribut objek yang dimiliki oleh suatu objek tidak dimiliki oleh objek lainnya. Artinya, antara dua objek tersebut berdasarkan himpunan atribut yang dimilikinya merupakan suatu relasi *inclusion* (*superset/subset*). Ini merupakan suatu *proper superset/subset*. Dalam paper ini, relasi demikian dinamakan sebagai relasi objek komposisi. Objek dengan himpunan objek yang merupakan *proper set* dinamakan objek komposisi, sedangkan objek yang merupakan *subset*-nya adalah elemen objek komposisi, contoh objek a dengan objek b (lihat pada Tabel 1) [1].

Objek-objek pada *dataset* K *hypergraph* H merupakan relasi *partially similar*, lihat Tabel 1, dan Tabel 2. Relasi objek antar objek a dan b merupakan relasi *partially similar*. Relasi objek antar tiga objek a, b, dan d merupakan relasi *partially similar*. Namun demikian, ada yang menarik relasi antara objek a dan b, keduanya membentuk suatu relasi yang dinamakan *inclusion relationship*. Kardinalitas dua objek tersebut berbeda, $|b| < |a|$. Objek $b = \{1, 2, 4\}$, dan objek $a = \{1, 2, 3, 4\}$. Objek b merupakan *proper subset* dari objek a berdasarkan *hyperedges* yang dimilikinya, $b \subset a$.

Persamaan dasar relasi objek komposisi dapat di lihat pada Persamaan (6). Objek v merupakan objek komposisi adalah superset bagi objek u. Objek u dinamakan elemen objek komposisi. Ada dua kemungkinan yaitu u identik dengan objek v, atau kemungkinan kedua objek u merupakan proper subset terhadap v, $d(v) > d(u)$.

$$\pi_v(u) = \begin{cases} v = u, \text{identical, symmmetri, } d(v) = d(u) \\ v \supset u, \text{composite, antisymmetri, } d(v) > d(u) \end{cases} \quad (6)$$

Pada koleksi tiga objek u, v, dan w, maka memiliki sejumlah kemungkinan relasi objek komposisi yang terbentuk, pada elemen objek komposisi yaitu: identik, *partially similar*, berbeda, atau *nested*. Pada relasi elemen objek yang *nested* memiliki sifat *transitive*. Ini merupakan *partially order set* (poset).

2.4. Algoritma Identifikasi dan Konstruksi Relasi Objek Transitive Closure

Identifikasi dan konstruksi relasi objek komposisi pada koleksi objek dapat di lihat di bawah ini. Tahapan proses identifikasi relasi objek adalah sebagai berikut. Algoritma ini akan menghasilkan umpan balik yaitu apakah koleksi objek hanya partially similar, ataukah ada sedikitnya dua objek yang membentuk relasi objek komposisi. Jika ada relasi objek komposisi maka algoritma ini akan menghasilkan sedikitnya satu graph/subgraph. Jika semua objek dalam koleksi objek terserap ke dalam satu graph, maka koleksi objek tersebut membentuk suatu *connected graph*. Jika hanya sebagian objek, maka akan membentuk sedikitnya dua *forest graphs*. *Forest graph* merupakan suatu *component graph*.

Algoritma:

Input: Himpunan objek $K, v_i \in K, d(v_i) < d(v_{i+1}), 1 \leq i < |K|$, dan himpunan hyperedges $E_K, e_i \in E_K, \delta(e_i) \geq \delta(e_{i+1}), 1 \leq i < |E_K|$

Output: Himpunan P, relasi objek komposisi yang *transitive closure*

1. Initialization $P = \emptyset$

2. **for each** vertex $v_i \in K, 1 \leq i \leq |K|$, as an eCandidate(v_i).
 - a. Initialization $T = \emptyset$ and $T' = \emptyset$
 - b. Insert v_i ke T'
 - c. Form T from first hyperedge, $T = e_1 - T', e_1 \in E_K$
 - d. **for each** hyperedge, $e_p \in E_K, 2 \leq p \leq |E_K|$ which incident with $v_i, h(v_i, e_p) = 1$
 - 1) Form $N = e_p - T'$
 - 2) **if** $T - N \neq \emptyset$ **then** delete $T - N$ from T
 - 3) **if** $T = \emptyset$ **then** access the next vertex v_{i+1} , do the step1
 - e. **if** $\delta(e_p) = 1$ **and** $v_i \in e_p$ **then** access the next vertex v_{i+1} , do the step1
 - f. **if** all e_p has been accessed **and** $T \neq \emptyset$ **then**
 - 1) makeComposition(T, v_i) = $\{(u, \{v_i\}) \mid \text{Composition}(u, W), u \in T, v_i \in W\}$
 - 2) Insert makeComposition(T, v_i) to P
3. Access the next vertex v_{i+1} , do the step1

Berikut ini contoh tahapan eksekusi algoritma identifikasi dan konstruksi relasi objek komposisi dalam bentuk *transitive closure*, lihat Tabel 3.

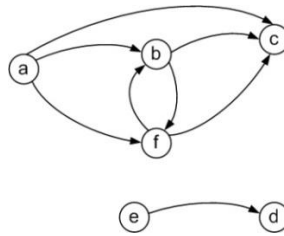
Tabel 3. Proses identifikasi Algoritma-2

eCandidate(c)	hyperedge 2: $\{a, b, c, d, e, f\}, T = \{a, b, d, e, f\}, T' = \{c\}$ hyperedge 1: $\{a, b, c, f\}, N = \{a, b, f\}, T - N = \{d, e\}$ is removed, then $T = \{a, b, f\}$ hyperedge 4, ignore: $\{a, b, e, f\}, h(c, 1) = 0, T = \{a, b, f\}$ hyperedge 5, ignore: $\{d, e\}, h(c, 1) = 0, T = \{a, b, f\}$ hyperedge 3, ignore: $\{a\}, h(c, 1) = 0, T = \{a, b, f\}$ $P = \{(a, \{c\}), (b, \{c\}), (f, \{c\})\}$
eCandidate(d)	hyperedge 2: $\{a, b, c, d, e, f\}, T = \{a, b, c, e, f\}, T' = \{d\}$ hyperedge 1, ignore: $\{a, b, c, f\}, h(d, 1) = 0, T = \{a, b, c, e, f\}$ hyperedge 4, ignore: $\{a, b, e, f\}, h(d, 4) = 0, T = \{a, b, c, e, f\}$ hyperedge 5: $\{d, e\}, N = \{e\}, T - N = \{a, b, c, f\}$ is removed, then $T = \{e\}$ hyperedge 3, ignore: $\{a\}, h(d, 3) = 0$ $P = \{(a, \{c\}), (b, \{c\}), (f, \{c\}), (e, \{d\})\}$

3. Hasil dan Pembahasan

Pada eksperimen ini, model hypergraph dan algoritma dikembangkan menggunakan C++ dan memanfaatkan STL (*Standard Template Library*). Eksperimen yang dilakukan terhadap koleksi objek $\{a, b, c, d, e, f\}$ pada Tabel 1, dan dari hasil eksperimen menunjukkan algoritma dapat mengidentifikasi dan mengkonstruksi relasi objek komposisi. Visualisasi dari relasi objek tersebut dapat di lihat pada Gambar 1. Hasil identifikasi yang dilakukan algoritma di atas (section 2.4) membentuk relasi *transitive closure*. Objek a merupakan proper superset terhadap objek b, c, dan f. Objek a memiliki *path* ke objek c melalui objek b, dan koneksi langsung objek a ke objek c adalah *transitive edge*. Objek b dan objek f membentuk relasi objek siklis, yaitu dua objek komposisi b dan f. Objek komposisi f memiliki elemen objek b, dan objek komposisi lainnya objek b memiliki elemen objek komposisi f. Objek b dan objek f, merupakan dua objek yang identik berdasarkan himpunan atribut yang menjadi deskripsi objek yang bersangkutan. Eksperimen juga dilakukan pada objek patung Durga, pembahasan lanjut dapat dilihat pada [1].

Pada paper ini, eksperimen dilanjutkan terhadap dummy data dengan objek sebanyak 1000, 1500, 2000, 2500, 3000, 3500, 4000, dan 5000, algoritma dapat berfungsi dengan baik. Waktu eksekusi pada *dataset dummy data* dapat dilihat pada Gambar 3. Objek-objek memiliki rentang atribut 15 hingga 52. Banyaknya *incident* yang terbentuk 17023 pada 1000 objek, dan 167376 *incidents* pada 5000 objek. Atribut (feature) yang merupakan deskripsi objek berupa pasangan <atribut, nilai-atribut> yang dinamakan *feature-value*, dan juga berupa atribut tanpa menyertakan nilai-atribut, dinamakan *feature-structure*. Objek-objek dummy dibentuk dari tiga objek dasar yang berbeda namun memiliki intersection di antara mereka. Dari tiga objek dasar tersebut dipilih secara acak untuk membentuk objek baru yang memungkinkan identik, berbeda, atau *partially similar*, secara acak. Seiring dengan bertambahnya objek yang terbentuk, maka pembentukan objek berikutnya dipilih secara acak berdasarkan banyaknya objek terakhir sehingga semakin banyak pilihan. Pembentukan objek baru diiringi juga dengan penambahan atribut baru, pengurangan atribut, atau menyalin penuh atribut yang dimiliki oleh suatu objek yang terpilih.



Gambar 2. Visualisasi relasi objek komposisi koleksi objek {a, b, c, d, e, f}



Gambar 3. Waktu eksekusi *dummy data* pada identifikasi berdasarkan (a) *feature-value*, (b) *feature-structure*

4. Simpulan

Model *Adjacency hyperedges matrix*, formulasi persamaan koneksi yaitu relasi objek, dan algoritma identifikasi objek komposisi dapat digunakan untuk melakukan identifikasi relasi objek komposisi, dan mengkonstruksi relasi *poset* berdasarkan himpunan attribute yang menjadi deskripsi objeknya yang memenuhi operator relasi *superset*, dan koneksi antar objek yang terbentuk merupakan suatu *poset* yang *transitive closure*. Hasil eksperimen menunjukkan pula model dapat menangani data yang lebih banyak dengan memanfaatkan STL.

Daftar Pustaka

- [1] Soetrisno, Hariadi, N., and Suhartanto, H., "Constructing Transitive Closure on Multigraph using Adjacency Hyperedges Matrix," 2017 International Conference on Advanced Computer Science and Information Systems, Jakarta, Indonesia, 28-29 Oktober 2017.
- [2] Cahya, S., "Identifikasi Partially Similar Objects Menggunakan Adjacency Hyperedges Matrix," Konferensi Nasional Sistem Informasi 2016, Batam Riau, Agustus 2016, pp. 624-629, ISBN: 978-602-74905-0-5.
- [3] S. Soetrisno, S., and H. Suhartanto, "Adjacency Hyperedges Matrix for Multi-Objects Connection Model on Non-uniform Object Features. *International Journal of Advancements Computing Technology (IJACT)*. Vol. 6, No. 6, 2014, pp. 113-125.
- [4] R.T. Olszewski, "Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data," Doctoral Philosophy Dissertation, School of Computer Science Carnegie Mellon University, Pittsburgh PA 15213. CMU-CS-01-108, 2001. pp. 1-23.
- [5] P. Ren, "Developments in Structural Learning Using Ihara Coefficients and Hypergraph Representation," Doctor of Philosophy (Thesis), 2010. Department of Computer Science, University of York.
- [6] L. Sun, S. Ji, and J. Ye, "Hypergraph Spectral Learning for Multi-label Classification," KDD'08. , August 24-27, 2008, Las Vegas, USA. ACM 978-1-60558-193-4/08/08.
- [7] M.O. Saliu, and G. Ruhe, G., "Bi-Objective Release Planning for Evolving Software Systems," *ESEC/FSE'07*, September 3-7, 2007, Cavtat near Dubrovnik, Croatia, *ACM SIGSOFT symposium on The foundations of software engineering*, ACM 978-1-59593-811-4/07/0009, pp. 105-114.
- [8] D. Zhou, J. Huang, and B. Scholkopf, "Learning with Hypergraph: Clustering, Classification, and Embedding," *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2006. Accessed from <http://research.microsoft.com/en-us/um/people/denzho/papers/hyper.pdf>.