Metode Temu Kembali Data Dengan Uji Similaritas Levenshtein Pada Sumber Data Terdistribusi

Romdhoni Susiloatmadja, I Made Wiryana, Adang Suhendra, Lintang Yuniar Banowosari dan Lily Wulandari
Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424
{romdhoni, mwiryana, adang, lintang, lily}@staff.gunadarma.ac.id

ABSTRAK

Temu kembali data pada sumber data terdistribusi dapat mengalami kendala jika terdapat ketidak konsistenan isi data, yaitu data yang sebenarnya sama, tetapi disimpan secara berbeda pada sumber data yang berbeda. Penelitian ini bertujuan mengembangkan metode temu kembali data untuk menemukan data yang disimpan secara tidak konsisten pada sumber data yang berbeda tanpa dilakukan penyeragaman atau perubahan data. Metode untuk temu kembali data yang tidak konsisten tersebut, pada penelitian ini data yang relevan dicari dari data yang sama atau mirip dengan data query pencarian. Kesamaan atau kemiripan data ini ditentukan berdasarkan hasil uji similaritas Levenshtein. Data yang nilai similaritasnya paling besar menunjukkan bahwa data tersebut paling relevan yang berarti merujuk pada data dari obyek yang sama. Metode yang dikembangkan pada penelitian ini berhasil menemukan data yang sama maupun data yang tidak konsisten yang masih memiliki banyak kemiripan dengan tetap mempertahankan keaslian data. Kelemahan metode yang dikembangkan ini yaitu tidak dapat menemukan ketidak konsistenan data yang hanya memiliki sedikit kemiripan dan data yang disimpan dengan istilah yang berbeda.

Kata Kunci : *data, sumber data, temu kembali, terdistribusi, tidak konsisten*

PENDAHULUAN

Sumber data terdistribusi adalah sumber data yang tersebar pada sejumlah lokasi. Setiap lokasi tersebut memiliki kewenangan sendiri dalam mengelola sumber data. Masing-masing lokasi bisa melakukan transaksi lokal maupun transaksi global. Setiap lokasi dapat mengelola sumber data masing-masing sesuai dengan kebutuhannya, sehingga berpeluang terjadinya ketidak konsistenan isi data pada lokasi yang satu dan yang lainnya. Ketidak konsistenan ini dapat menyebabkan terjadinya kesulitan dalam melakukan integrasi atau pertukaran data.

Diperlukan suatu metode untuk mengenali data dari sumber data yang terdistribusi apakah data tersebut mengacu pada obyek yang sama atau berbeda. Misalnya data dari dua orang yang berbeda tetapi memiliki nama yang sama, atau sebaliknya untuk dua nama yang disimpan secara berbeda tetapi sebetulnya adalah data dari orang yang sama. Pada basis data relasional terdapat atribut kunci relasi antar data, yaitu berdasarkan *primary key* dan *foregn key*, sehingga temu kembali data dapat dilakukan berdasarkan kunci relasional tersebut. Berdasarkan kunci relasional yang sama, dapat ditemukan data meskipun atribut data yang lain disimpan secara tidak konsisten. Hal ini tidak dapat dilakukan jika pada sumber data terdistribusi tidak terdapat atribut kunci relasional. Untuk memecahkan persoalan tersebut maka perlu metode untuk menentukan data mana yang memiliki relevansi dengan kesamaan obyek sehingga kesamaan data tersebut lebih diprioritaskan.

Tujuan penelitian ini adalah mengembangkan metode untuk menemukan data dari obyek yang sama tetapi data tersebut disimpan secara berbeda pada sumber data yang berbeda dan pada sumber data tersebut tidak terdapat atribut kunci relasional. Kontribusi hasil penelitian ini adalah metode temu kembali data dari obyek yang sama tetapi disimpan secara tidak konsisten pada sumber data yang terdistribusi dengan menggunakan uji similaritas *Levenshtein*.

Sumber Data Terdisribusi

Sumber data terstruktur merupakan basis data, yaitu kumpulan data yang saling berhubungan yang dirancang untuk

memenuhi kebutuhan informasi dari suatu organisasi [1]. Pemanfaatan sumber data adalah untuk dapat memenuhi sejumlah tujuan, yaitu: kecepatan dan kemudahan, efisensi ruang penyimpanan, keakuratan, ketersediaan, kelengkapan, keamanan, dan kebersamaan pemakaian [2]. Alasan dilakukan penyimpanan data secara terdistribusi dalam suatu sistem, semata-mata adalah dengan pertimbangan performa [3].

Sebuah sistem dengan sumber data terdistribusi terdiri dari kumpulan sejumlah *site*. Masing-masing *site* tersebut dapat berpartisipasi dalam pemrosesan transaksi mengakses data pada suatu *site* atau beberapa *site* [4]. Sistem sumber data terdistribusi terdiri dari sekumpulan lokasi yang dihubungkan bersama-sama melalui beberapa jenis jaringan komunikasi, dimana setiap lokasi lengkap dengan sistem sumber data tersendiri di dalamnya, tetapi semua lokasi telah sepakat untuk bekerja sama sehingga pemakai pada suatu lokasi dapat mengakses data dimanapun dalam jaringan tersebut persis seperti jika data disimpan dalam lokasi pemakai itu sendiri [5].

Data diperlukan untuk pengambilan keputusan. Oleh karena itu kebenaran data sangat penting untuk menghindari kesimpulan yang salah. Masalah kualitas data pada sumber data, dapat terjadi karena salah eja saat entri data, informasi hilang, atau data tidak valid lainnya. Masalah kualitas data secara garis besar dibedakan menjadi masalah sumber tunggal dan masalah multi sumber. Pada kedua sumber tersebut terdapat masalah pada tingkat skema dan masalah pada tingkat *instance*. Masalah pada tingkat skema juga tercermin pada tingkat *instance*. Masalah di tingkat skema dapat ditangani dengan desain skema yang lebih baik (skema evolusi), dan skema integrasi. Masalah di tingkat *instance* mengacu pada kesalahan dan ketidak konsistenan dalam isi data aktual yang tidak terlihat pada level skema. Permasalahan tersebut adalah fokus utama pada pembersihan data. Masalah pada sumber tunggal kemungkinan dapat meningkat pada masalah multi sumber [6].

Penelitian yang berkaitan dengan skema dan integrasi skema, sejumlah peneliti memusatkan perhatian pada masalah dupliksi, identifikasi dan eliminasi, antara lain dilakukan oleh *Galhardas, H*., dkk. [7], *Hernandez, M.A*.; *Stolfo* [8], *Lee, M.L*., dkk. [9]. Beberapa peneliti berkonsentrasi pada masalah umum tidak terbatas namun relevan dengan pembersihan data, seperti pendekatan data mining khusus, antara lain dilakukan oleh *Savasere, A*., dkk. [10], dan dengan pendekatan transformasi data berdasarkan pencocokan skema seperti yang dilakukan oleh *Milo, T*. dan *Zohar, S*. [11]. Beberapa penelitian mengusulkan dan menyelidiki penanganan pembersihan data yang lebih komprehensif dan seragam yang mencakup beberapa fase transformasi, operator tertentu dan implementasinya [9].

Metode untuk rekonsiliasi data pada sumber data terdistribusi yang digunakan oleh *Champlin*, dkk [12], yaitu untuk penggabungan dua tabel data, yaitu tabel data pertama dan tabel data kedua, kunci gabung dua tabel data tersebut adalah satu atau lebih atribut pada tabel data pertama dan satu atau lebih atribut pada tabel data kedua. Data yang disimpan secara tidak konsisten pada atribut kunci gabung dua tabel data tersebut diatasi dengan cara memodifikasi data, yaitu data dipetakan ke skema baru dalam beberapa atribut baru yang ditentukan, dan dilakukan proses pembersihan data dengan mengikuti aturan yang ditentukan. Metode ini memiliki kelebihan yaitu dapat menemukan data yang disimpan secara tidak konsisten. Namun demikian metode ini masih mempunyai kelemehan dan keterbatasan yang antara lain adalah dilakukan modifikasi data sehingga data menjadi tidak sama seperti data aslinya.

Pada metode yang diusulkan dalam penelitian ini tidak dilakukan perubahan terhadap data, sehingga data tetap dipertahankan sebagaimana aslinya. Penelitian ini menggunakan pendekatan *fuzzy query* berdasarkan kondisi *Generalised Logical Condition* (*GLC*). Formula *GLC* pada awalnya dibuat mampu menangkap ekspresi linguistik ke bagian *Where* dari *Structured Query Language* (*SQL*). Pengguna mencari sumber data untuk mendapatkan data yang dibutuhkan untuk analisis, membuat keputusan atau untuk memuaskan keingintahuan mereka. *SQL*

adalah bahasa query standar untuk sumber data relasional. GLC untuk bagian *Where* dari *SQL* diciptakan berdasarkan ekspresi linguistik [13]. Untuk pembacaan lebih lanjut, penting untuk menentukan *Query Compatibility Index* (*QCI*). *QCI* digunakan untuk menunjukkan bagaimana catatan yang dipilih memenuhi kriteria *query*. *QCI* memiliki nilai dari interval [0, 1] dengan arti sebagai berikut: nilai 0 berarti *record* tidak memenuhi *query*, nilai 1 berarti *record* sepenuhnya memenuhi *query*, dan nilai interval antara 0 sampai dengan 1 berarti sebagian memenuhi *query* dengan jarak ke kepuasan *query* penuh.

SQL digunakan untuk memperoleh data dari sumber data relasional. Perbaikan *fuzzy query SQL* memiliki kelebihan dalam kasus ketika pengguna tidak dapat secara jelas menentukan kriteria seleksi atau bila pengguna ingin memeriksa data yang hampir memenuhi kriteria yang diberikan. Penelitian yang dilakukan *Hudec* [14] adalah memeriksa realisasi konsep *fuzzy query*. Untuk tujuan ini, dibuat kondisi logis *general fuzzy* untuk instruksi "*Where*" yang merupakan bagian dari *SQL*. Hal ini memungkinkan pengguna untuk membuat *query* dengan istilah linguistik. Model yang diusulkan adalah perpanjangan dari *SQL*, sehingga tidak harus dilakukan modifikasi di dalam sumber data.

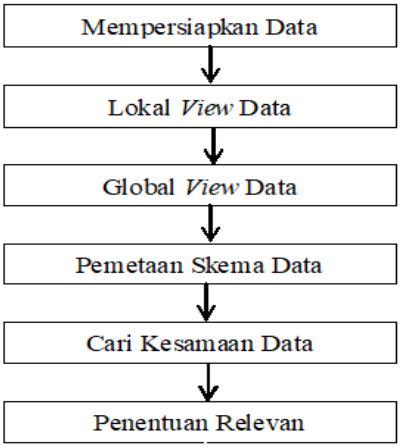
Uji Similaritas *Levenshtein*

Uji similaritas *Levenshtein* yang pertama kali dikemukakan oleh *Vledimir Levenshtein* [15] adalah cara yang digunakan untuk mengukur nilai kesamaan atau kemiripan antara dua buah kata atau *string*, yang diperoleh dengan mencari cara termudah atau operasi edit yang paling sedikit untuk memodifikasi agar dua string menjadi sama. Operasi memodifikasi string untuk keperluan ini adalah memasukkan karakter ke dalam string, menghapus karakter dari suatu string, dan mengganti karakter string dengan karakter lain. Similaritas atau nilai kemiripan dihitung berdasarkan perbandingan antara jumlah operasi edit yang paling sedikit dan nilai string terpanjang.

Berdasarkan hasil survei terhadap beberapa uji similaritas yang dilakukan oleh *Wael H. Gomaa* dan *Aly A. Fahmy* [16], uji similaritas *Levenshtein* memilki kelebihan, yaitu cocok digunakan untuk uji similaritas string berdasarkan karakter seperti yang terdapat pada ketidak konsistenan data yang digunakan dalam penelitian ini, yaitu terdapat perbedaan dalam penulisan karena salah ketik atau salah eja. Sedangkan uji similaritas yang lain lebih cocok untuk kondisi yang berbeda, misalnya *N-gram* untuk membandingkan masing-masing karakter atau kata dalam dua string, *Needleman-Wunsch* cocok untuk dua kata yang memunyai panjang kesaman dan derajat kesamaan yang berbeda, *Jaro* cocok untuk similaritas berdasarkan pada jumlah dan urutan karakter umum antara dua string, *Jaro-Winkler* lebih menguntungkan untuk string yang sesuai, *Longest Common SubString (LCS)* cocok untuk uji similaritas berdasarkan panjang dari rantai karakter yang berdekatan yang ada pada kedua string.

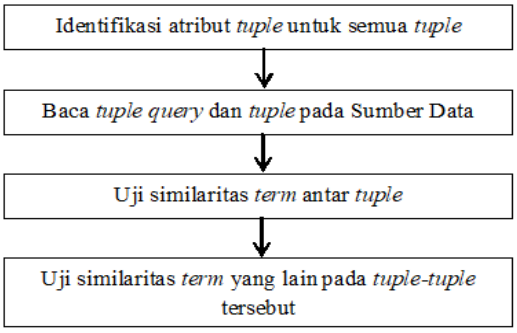
METODE PENELITIAN

Langkah-langkah atau tahapan yang dilakukan untuk mengembangkan metode temu kembali data dengan uji similaritas *Levenshtein* dalam penelitian ini adalah seperti ditunjukkan pada Gambar 1, yaitu meliputi persiapan data yang akan digunakan untuk penelitian ini, pemeriksaan bagaimana isi data pada setiap sumber data (*local view data*), pemeriksaan bagaimana isi data pada keseluruhan sumber data (*global view data*), pemetaan skema data, pencarian kesamaan atau kemiripan data antara data pada sumber data dan data *query* pencarian, penentuan data yang relevan antara data yang dicari pada sumber data dan data *query* pencarian.



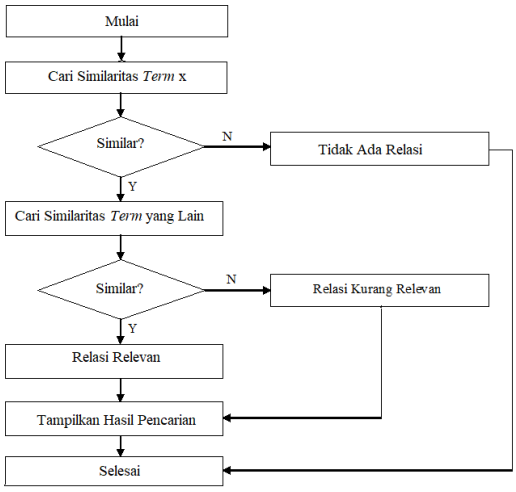
Gambar 1. *Tahapan Penelitian*

Secara umum, pencarian kesamaan atau kemiripan data antara data pada sumber data dan data *query* pencarian adalah berdasarkan similaritas data. Pencarian kesamaan atau kemiripan data ini dilakukan pada setiap sumber data yang dikehendaki, yaitu minimal pada dua sumber data yang dikehendaki. Langkah-langkah pencarian ini adalah seperti ditunjukkan pada Gambar 2, yaitu meliputi identifikasi atribut data untuk semua data pada setiap sumber data, pembacaan data *query* pencarian dan data yang dicari pada sumber data yang akan diuji similaritasnya, pencarian kesamaan atau kemiripan suatu atribut pada data yang diuji dengan menggunakan uji similaritas *Levenshtein*, dan pencarian kesamaan atau kemiripan atribut yang lainnya pada data-data yang diuji tersebut.



Gambar 2. *Tahapan Pencarian Kesamaan Data*

Kesamaan atau kemiripan antar data ditentukan berdasarkan similaritas atribut-atribut pada data-data yang diuji tersebut dengan menggunakan uji similaritas *Levenshtein*. Atribut-atribut dikatakan sama jika similaritasnya 100% atau 1 dan dikatakan mirip jika similaritasnya mendekati 100% atau mendekati 1. Jika tidak ada kesamaan atau kemiripan atribut berarti data yang diuji tidak relevan dengan data *query* pencarian. Langkah selanjutnya adalah menguji kesamaan atau kemiripan untuk data yang lain pada sumber data yang lain. Jika ada kesamaan atau kemiripan atribut berarti ada relasi antara data-data yang diuji tersebut. Masih pada data-data yang diuji ini, diulang lagi langkah-langkah tersebut untuk pencarian kesamaan atau kemiripan pada atribut-atribut yang lainnya. Jika ada atribut lain yang sama atau mirip berarti relasi kedua data tersebut relevan. Selanjutnya data-data yang relevan ditampilkan sebagai hasil pencarian yang ditemukan. Data-data yang tidak relevan tidak ditampilkan atau bukan sebagai hasil pencarian. Proses pencarian data yang relevan adalah seperti ditunjukkan pada Gambar 3.



Gambar 3. *Pencarian Data yang Relevan*

HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini adalah data yang dikumpulkan dari berbagai organisasi atau bagian yang saling terpisah pada sebuah institusi pendidikan. Masing-masing bagian memiliki dan mengelola data sesuai dengan kebutuhan organisasinya. Pada prakteknya, antar bagian juga saling berbagi atau bertukar data. Berdasarkan hasil dari tahap awal penelitian yang ditunjukkan pada

Gambar 1, yaitu tahap *local view data*, tahap *global view data*, dan tahap pemetaan skema data, maka ditentukan atribut-atribut data yang dapat digunakan untuk pengembangan metode temu kembali data pada penelitian ini, yaitu seperti diperlihatkan pada Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1. *Tabel Personalia*

KD	NAMA DOSEN	BAGIAN
1234	Romdhoni Susiloatmadja	D3MI
2345	Budi Santoso	WD Kom
3456	Budi Santosa	LB

Tabel 2. *Tabel Keuangan*

KD	NAMA DOSEN
1234	Romdhony Susiloatmaja
2345	Budi Santoso N
3456	Budi Santoso

Tabel 3. *Tabel Jadwal*

NAMA DOSEN	MATA KULIAH
Romdoni S	Fisika dan Kimia
Budi Santoso	Komunikasi
Budi Santoso	Bahasa Inggris

Pada Tabel 1 dan Tabel 2 terdapat atribut KD yang berisi kode unik. Atribut KD ini dapat dijadikan sebagai atribut kunci relasional antara Tabel 1 dan Tabel 2. Sementara itu, pada Tabel 3 tidak terdapat atribut KD tersebut, sehingga tidak terdapat atribut kunci relasional antara Tabel 3 dengan Tabel 1 maupun dengan Tabel 2.

Terdapat ketidak konsistenan isi data pada sumber data di bagian yang satu dan lainnya seperti yang diperlihatkan pada Tabel 1, Tabel 2, dan Tabel 3, yaitu terdapat data dari seorang dosen yang disimpan secara tidak konsisten. Contohnya, pada Tabel 1 terdapat data dosen yang disimpan dengan nama "Romdhoni Susiloatmadja", sedangkan pada Tabel 2 data tersebut disimpan dengan nama "Romdhony Susiloatmaja" dan pada Tabel 3 data tersebut disimpan dengan nama "Romdhoni S". Temu kembali data dari dosen tersebut pada Tabel 1 dan Tabel 2 dapat dilakukan dengan menggunakan kode kunci relasional yaitu "1234". Dengan menggunakan *query* pencarian "1234" maka ditemukan data pada Tabel 1 dengan nama "Romdhoni Susiloatmadja", dan pada Tabel 2 dengan nama "Romdhony Susiloatmaja". Tetapi dengan *query* pencarian tersebut tidak ditemukan data pada Tabel 3. Demikian juga data untuk dosen yang lainnya tidak dapat ditemukan pada Tabel 3 dengan menggunakan *query* pencarian yang berupa kode kunci relasional, yaitu isi data pada atribut KD.

Pada Tabel 3 terdapat data dari dua orang dosen yang berbeda, tetapi disimpan dengan nama yang sama, yaitu "Budi Santoso", sedangkan pada Tabel 1 dan Tabel 2 data kedua orang dosen tersebut disimpan dengan nama yang berbeda, yaitu pada Tabel 1 disimpan dengan nama "Budi Santoso" dan "Budi Santosa", dan pada Tabel 2 disimpan dengan nama "Budi Santoso N" dan "Budi Santoso". Dengan menggunakan *query* pencarian "Budi Santoso", maka ditemukan data dari kedua orang dosen tersebut pada Tabel 2 dan Tabel 3. Sementara itu, yang diharapkan adalah data dari salah satu dosen saja. Solusi yang ditawarkan pada penelitian ini yaitu dengan menggunakan uji similaritas *Levenshtein* untuk menemukan data yang paling cocok dengan *query* pencarian. Cara mencari data yang paling cocok dengan menggunakan uji similaritas *Levenshtein* ini adalah menggunakan tahapan seperti ditunjukkan pada Gambar 2.

Nilai hasil uji similaritas adalah antara 0% atau 0 sampai dengan 100% atau 1. Semakin besar nilai similaritas, maka data dikatakan semakin mirip. Jika similaritas 100% atau 1 maka dikatakan data yang diuji sama dengan data *query* pencarian. Data yang paling relevan adalah data dengan nilai similaritas paling besar. Namun demikian data lain dengan nilai similaritas yang lebih kecil dan masih diatas ambang batas minimal tetap ditampilkan pada hasil pencarian. Hal ini dimaksudkan sebagai kontrol untuk mencocokkan apakah data yang ditemukan tersebut betul atau salah.

Data yang ditampilkan sebagai hasil pencarian yang ditemukan adalah data yang relevan, yaitu data yang sama atau mirip

dengan *query* pencarian. Oleh karena itu diperlukan batasan atau nilai ambang untuk menentukan data yang relevan. Berdasarkan hasil penelitian ini ditentukan batasan atau nilai ambang tersebut yaitu dengan nilai similaritas minimal 60% atau 0,6. Data dengan nilai similaritas dibawah 60% atau 0,6 dianggap sebagai data yang tidak relevan sehingga tidak ditampilkan pada hasil pencarian.

Jika batasan atau nilai ambang untuk data yang relevan dan ditampilkan sebagai hasil pencarian yang ditemukan terlalu besar, maka akan banyak data yang sebenarnya relevan tetapi ternyata tidak ditampilkan karena nilai similaritasnya dibawah nilai ambang. Demikian juga sebaliknya, yaitu jika batasan atau nilai ambang tersebut terlalu kecil, maka akan terlalu banyak data yang ditampilkan termasuk data yang sebenarnya tidak relevan dan seharusnya tidak ditampilkan.

Hasil pada penelitian ini, dapat ditemukan data yang sama atau data yang disimpan secara konsisten. Misalnya pada salah satu tabel data disimpan dengan nama "Budi Santoso" dan pada tabel data yang lain juga disimpan dengan nama "Budi Santoso". Pada penelitian ini juga dapat ditemukan data yang disimpan secara tidak konsisten dengan banyak kemiripan atau hanya ada sedikit perbedaan dalam penulisan atau ejaan. Misalnya pada salah satu tabel data disimpan dengan nama "Budi Santosa" dan pada tabel data yang lain disimpan dengan nama "Budi Santoso". Contoh yang lain, misalnya pada salah satu tabel data disimpan dengan nama "Romdhoni Susiloatmadja" dan pada tabel data yang lain disimpan dengan nama "Romdhony Susiloatmaja".

Metode temu kembali data dengan uji similaritas *Levenshtein* yang dikembangkan dalam penelitian ini dapat menemukan data yang disimpan secara tidak konsisten pada sumber data yang berbeda atau pada sumber data yang terdistribusi tanpa dilakukan penyeragaman atau perubahan data pada sumber data, sehingga data pada setiap sumber data tetap dipertahankan sebagaimana aslinya. Hal ini berbeda dengan metode lain yang disebutkan pada penulisan ini, yaitu pada metode lain tersebut dilakukan perubahan atau modifikasi pada data seperti pada metode yang digunakan oleh *Lee, M.L*., dkk. [9], *Savasere, A*., dkk. [10], *Milo, T.* dan *Zohar, S*. [11], dan *Champlin*, dkk [12].

Hasil pada penelitian ini, untuk ketidak konsistenan data seperti pada Tabel 1 dengan nama "Romdhoni Susiloatmadja" dan pada Tabel 3 dengan nama "Romdoni S" ternyata metode temu kembali data dengan uji similaritas *Levenshtein* ini kurang cocok. Untuk data dari dosen ini, jika dilakukan pencarian dengan *query* "Romdhoni Susiloatmadja", maka hasil pencarian pada Tabel 3 tidak menampilkan data "Romdoni S" karena nilai similaritasnya kurang dari 60% atau 0,6. Demikian pula jika dilakukan pencarian dengan *query* "Romdoni S" atau dengan *query* "Romdhoni", maka hasil pencarian pada Tabel 1 juga tidak menampilkan data "Romdhoni Susiloatmadja" karena nilai similaritasnya kurang dari 60% atau 0,6. Dengan demikian, berdasarkan batasan nilai similaritas minimal 60% atau 0,6 data dengan nama "Romdhoni Susiloatmadja" dianggap tidak relevan dengan data dengan nama "Romdoni S" meskipun sebenarnya adalah data dari orang yang sama.

Metode temu kembali data dengan uji similaritas *Levenshtein* yang dikembangkan dalam penelitian ini tidak dapat menemukan data yang tidak konsisten karena data disimpan dengan nama inisial, misalnya pada salah satu tabel data disimpan dengan nama "Romdhoni Susiloatmadja" dan pada tabel data lainnya disimpan dengan nama "RS". Metode ini juga tidak dapat menemukan data yang tidak konsisten karena data disimpan dengan nama lain atau nama alias, atau data disimpan dengan istilah yang berbeda, misalnya pada salah satu tabel data disimpan dengan nama "Budi Santoso" dan pada tabel data lainnya disimpan dengan nama "Pak Kumis". Contoh yang lain misalnya pada salah satu tabel data disimpan dengan nama "Bulu Tangkis" dan pada tabel data lainnya disimpan dengan nama "Batminton". Jadi pada penelitian ini tidak dapat menemukan data yang disimpan secara tidak konsisten dengan sedikit kemiripan atau sama sekali tidak ada kemiripan.

PENUTUP

Pada penelitian ini dikembangkan metode temu kembali data pada koleksi data pada sumber data terdistribusi pada lingkungan yang heterogen yang terdapat ketidak konsistenan isi data. Heterogenitas data terjadi karena setiap sumber data memiliki otoritas mengelola data sesuai dengan kebutuhannya. Pada metode yang dikembangkan dalam penelitian ini, pencarian data pada sumber data terdistribusi tersebut menggunakan uji similaritas *Lavenshtein*. Data yang ditemukan adalah data yang memiliki nilai similaritas paling besar, yaitu sama dengan atau mendekati 100% atau 1. Metode yang dikembangkan dalam penelitian ini dapat menemukan data dari sebuah obyek yang disimpan secara konsisten maupun secara tidak konsisten pada sumber data yang berbeda atau terdistribusi dengan tetap mempertahankan keaslian data. Uji similaritas *Lavenshtein* pada metode yang dikembangkan dalam penelitian ini cocok untuk ketidak konsistenan data karena adanya perbedaan huruf atau ejaan dalam penulisan seperti yang terjadi pada penulisan yang salah ketik.

Kelemahan metode yang dikembangkan dalam penelitian ini yaitu tidak dapat menemukan data yang disimpan secara tidak konsisten dengan hanya sedikit kemiripan data, atau data dituliskan dengan nama lain atau dengan istilah yang berbeda. Untuk mengatasi hal ini, maka metode ini dapat dikembangkan lebih lanjut dengan memodifikasi uji similaritas yang digunakan atau dengan menggunakan uji similaritas yang lain.

DAFTAR PUSTAKA

[1] Connolly, Thomas M., and Carolyn E. Begg. Database *Systems : A Practical Approach to Design, Implementation, and Management*, Third Edition. Addison-Wesley, Reading, Massachusetts. 2002.

[2] Fathansyah, *Basis Data*, CV. Informatika Bandung, 2002.

[3] Ramakrishnan, R., Gehrke, J., *Database Management Systems*, Third Edition, TheMcGrawHill Companies, California, 2003.

[4] Korth H. F., Silberschatz A., *Database System Concept*, Fifth Edition, McGraw Hill Inc, USA, 2005.

[5] Date, C.J., *An Introduction to* Database *System*, Seventh Edition, Addison – Wesley Publishing Company, New York, 2000.

[6] Rahm, E. and Do, H. H., "Data Cleaning: Problems and Current Approaches". *IEEE Data Eng. Bull*., 23(4): 3-13, 2000.

[7] Galhardas, H.; Florescu, D.; Shasha, D.; Simon, E., "AJAX: An Extensible Data Cleaning Tool", *Proc. ACM SIGMOD Conf*., p. 590, 2000.

[8] Hernandez, M.A.; Stolfo, S.J., "Real-World Data is Dirty: Data Cleansing and The Merge/Purge Problem*", Data Mining and Knowledge Discovery*, 2(1):9-37, 1998.

[9] Lee, M.L.; Lu, H.; Ling, T.W.; Ko, Y.T., "Cleansing Data for Mining and Warehousing*", Proc. 10th Intl. Conf. Database and Expert Systems Applications (DEXA)*, 1999.

[10] Savasere, A.; Omiecinski, E.; Navathe, S., "An Efficient Algorithm for Mining Association Rules in Large Databases", *Proc. 21st VLDB*, 1995.

[11] Milo, T.; Zohar, S., "Using Schema Matching to Simplify Heterogeneous Data Translation", *Proc. 24th VLDB*, 1998.

[12] Champlin, et.al., *Fuzzy Join Key*, United States Patent, No. US 10,140,337 B2, Nov.27, 2018.

[13] Hudec, M., "An Approach to Fuzzy Sumber data Querying, Analysis and Realisation", *Computer Science and Information Systems*, 6(2): 127-140, 2009.

[14] Hudec, M., "Fuzzy Improvement of the SQL", *Yugoslav Journal of Operations Research* No. 2: 239-251, 2011.

[15] Vledimir Levenshtein, "Binary Codes Capable of Correcting Spurious Insertions and Deletions of Ones", *Probl. Inf. Transmission 1*, 8–17, 1965.

[16] Wael H. Gomaa and Aly A. Fahmy, "A Survey of Text Similarity Approaches", *International Journal of Computer Applications (0975 – 8887)*, 68(13): 13-18, 2013.