Lossless Data Deduplication: Alternatif Solusi untuk Mengatasi *Duplicated Record*

Ardijan Handijono

Jurusan Akuntansi S1, Fakultas Ekonomi, Universitas Pamulang, Tangerang Selatan, Banten, Indonesia
e-mail: dosen00853@unpam.ac.id

Submitted Date: January 22nd, 2020 Reviewed Date: January 25th, 2020
Revised Date: January 31st, 2020 Accepted Date: January 31st, 2020

Abstract

The implications of poor data quality bring negative effects for organisation through: increased operational costs, inefficient decision-making processes, lower performance and decreased both employee and customer satisfaction. Generally duplicated records can be handled by elimination or merge, but when duplicated records are occur in master table and used in a transaction the handling becomes not easy. This paper seeks to provide data deduplication solutions without losing the historical value of the transaction. To save all duplicated records which related to the transactions we use mapping table between Dimension table and facs table. Using this approach the quality of Dimension table increased since for this handling duplicated records process include enrichment process and delete dirty records and all dupplicated records which related to the transactions can be access completely in data warehouse, no transaction data loss.

Keywords: Duplicated Records, Deduplication, Data Cleansing, Supervised Duplicated Records Identification, Data Warehouse

Abstrak

Implikasi dari kualitas data yang buruk membawa dampak negatif bagi organisasi melalui: Meningkatnya biaya operasional, proses pengambilan keputusan yang tidak efisien, kinerja yang lebih rendah dan penurunan kepuasan karyawan dan pelanggan. Umumnya *duplicated records* dapat ditangani dengan eliminasi atau penggabungan, tetapi ketika *duplicated records* terjadi pada table master dan telah digunakan dalam transaksi penanganan menjadi tidak mudah. Makalah ini berupaya memberikan solusi deduplikasi data tanpa kehilangan nilai historis transaksi. Untuk menyimpan semua duplicated records yang terkait dengan suatu transaksi kami menggunakan tabel pemetaan antara tabel Dimensi dan tabel Facs. Dengan pendekatan ini kualitas tabel Dimensi meningkat karena pada proses penanganan duplicated records ini termasuk proses pengayaan dan menghapus record-record kotor dan semua duplicated records yang terkait dengan transaksi dapat diakses sepenuhnya di Data Warehouse, tidak ada data transaksi yang hilang.

Kata kunci: Duplikat Record, Deduplikasi, Pembersihan Data, Identifikasi Supervised Duplicated Records, Data Warehouse

1 Pendahuluan

Kualitas Data yang rendah memicu bertambahnya biaya operasional untuk mengidentifikasi dan memperbaiki kesalahan yang timbul. (Haug, Zachariassen, & Van Liempd, 2011). Banyak proyek-proyek Business intelliggnece (BI) yang gagal karena banyak nya Data kotor, problem data kotor akan semakin meningkat saat data diintegrasikan dari banyak sumber data (Bajpai & Metkewar, 2016). Duplicated record telah menjadi masalah besar pada data management (S. A. Babu, 2017). jadi menjadi sangat penting jika keputusan-keputusan bisnis bersandar pada Data yang bersih (Marsh, 2005). Suatu record diidentifikasi *duplicate* jika record tersebut merepresentasikan objek yang sama pada dunia nyata. (Skandar, Rehman, & Anjum, 2015). Solusi secara umum solusi pada *duplicate record* adalah *elimination* (Tamilselvi & Gifta, 2011; Tamilselvi & Saravanan, 2009) atau *merging (D. Elkington, Zeng, & Morris, 2016; Ker, VAISHNAV, & Dvinov, 2017)*, namun

untuk master data dalam sistem transaksional ditemukan masalah mendasar karena masing-masing *duplicated record* tersebut telah mempunyai nilai bisnis pada sejarah transaksi. Maka jika dilakukan *elimination* atau *merging* maka akan ada data transaksi yang hilang.

Aplikasi kecil perlu dikembangkan untuk menyederhanakan dan mempercepat proses identifikasi dan menyediakan banyak pilihan untuk menangani *duplicated record*. Data bersih dari aplikasi akan disimpan pada *maping table*.*mapping table* akan menyimpan semua *duplicated records* yang sudah valid diidentifikasi, dengan relasi ke satu record Master yang disimpan

dalam tabel Dimensi. Implementasi pendekatan ini akan berdampak pada proses *Surrogate Key Pipeline*, dalam aplikasi ETL (*Extract-Transform-Loading*), terutama dalam proses *Loading* ke Data warehouse.

2 Studi Pustaka

Bagian ini merupakan latar belakang materi yang diperlukan untuk bisa memahami permasalahan dan solusi yang akan dipaparkan pada makalah ini.

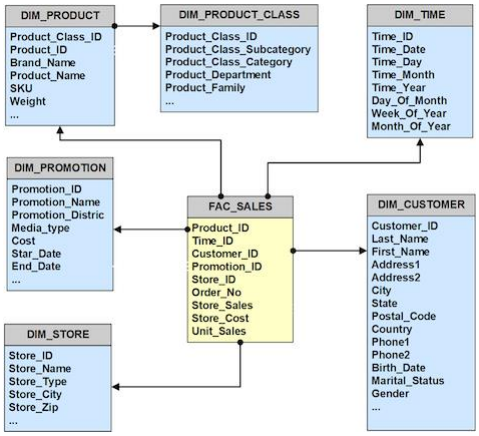
2.1 *Business Intelligence* dan Data *Warehouse*

Menurut (K. Babu, 2012) *Business Intelligence* (BI) adalah kategori yang luas pada aplikasi dan teknologi yang digunakan untuk mengumpulkan, menyimpan, menyediakan akses, menganalisa, dan membagikan informasi yang diperlukan oleh pengguna di seluruh perusahaan untuk meningkatkan kualitas pengambilan keputusan. Sedangkan Data Warehouse merupakan komponen utama pada BI, *Data Warehouse* fokus pada teknik penyimpanan dan penarikan data yang sangat besar. (Fleckenstein & Fellows, 2018)

Dengan BI memungkinkan perusahaan untuk mendapatkan informasi yang diperlukan dalam membuat keputusan bisnis yang sekaligus sebagai modal kekuatan daya saing. Tujuan utama BI adalah untuk meningkatkan kecepatan dan kualitas informasi. (K. Babu, 2012)

2.2 *Dimensional Modelling* pada Data Warehouse

Menurut (Kimball & Ross, 2013) *Dimensional modelling* yang digunakan pada *Data Warehouse* adalah implementasi RDBMS yang mengacu pada *Star-Schemas*. Disebut demikian karena hubungan antar tabel mirip dengan struktur sebuah bintang, lihat Gambar 1. Pada *Star-Schemas* hanya ada dua jenis table yaitu *Dimension Tables* dan *Fact Table*.



Gambar 1 Contoh Star-Schema

Dimension Tables **-** berisi data text yang berhubungan dengan kontek dari ukuran suatu proses bisnis. Semua *Dimension Tables* mempunyai *Primary Key* bertipe *integer* dan bersifat unik, yang disebut dengan *Surrogate Key*. Selain itu ada *Natual ID* yaitu *ID* yang berisi kode unik dari system transaksional, dan diikuti berapa *field attribute*.

Fact Tables **–** menyimpan suatu ukuran yang menunjukkan performan suatu proses bisnis. Semua *fact Table* juga menyimpan satu atau beberapa ukuran numerik yang disebut *Facts*. *Fact Tables* mempunyai beberapa *Foreign Keys* yang terhubung ke *Dimension table* terkait, untuk memberikan kontek dari ukuran yang ada pada *Fact Table*.

2.3 Menghubungkan *Facts* dan *Dimensions* untuk membentuk sebuah *Star-Schema*

Setiap proses bisnis dapat direpresentasikan dalam *dimensional modelling* yang terdiri dari sebuah *Fact table* yang berisi ukuran numerik dan dikelilingi beberapa *Dimension tables* yang berisi context Text yang memberikan penjelasan saat peristiwa tersebut terjadi.

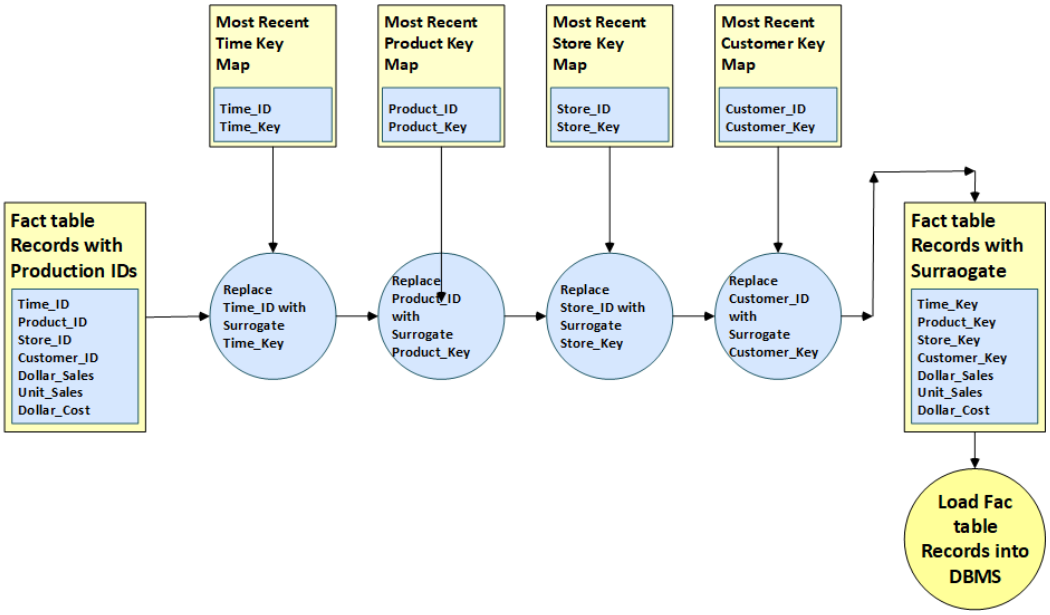
2.4 *Surrogate Key Pipeline*

Langkah terakir dalam proses ETL saat membentuk *Fact table* adalah menkonversikan semua *Natural ID* dari record yang masuk menjadi *Surrogate Keys* yang sesuai, seperti ditunjukkan pada Gambar 3. *Surrogate Key* ini diambil dari *Dimension Table* berdasarkan *Natural ID* dari record yang masuk, lalu

Surrogate Key yang didapatkan disimpan sebagai *Foreign Key* pada *Fact Table*. (Kimball & Caserta, 2011)

2.5 *Slowly Changing Dimension*

Menurut (Santos & Belo, 2011) ketika ada perubahan data pada ***Dimension table***, maka ada tiga respons dasar yang bisa dipilih, yaitu ***Slowly Changing Dimension*** (SCD) Tipe-1, Tipe-2, dan Tipe-3, namun menurut (Santos & Belo, 2011) baru-baru ini beberapa jenis tambahan telah diidentifikasi dan dapat digunakan sebagai alternatif atau sebagai kombinasi dari yang udah ada sebelumnya, yaitu SCD Tipe-4 dan Tipe-6 dan Tipe-0. Semua tipe tersebut beserta kelebihan dan kekurangannya masing-masing dapat di ringkas dalam Tabel 1.

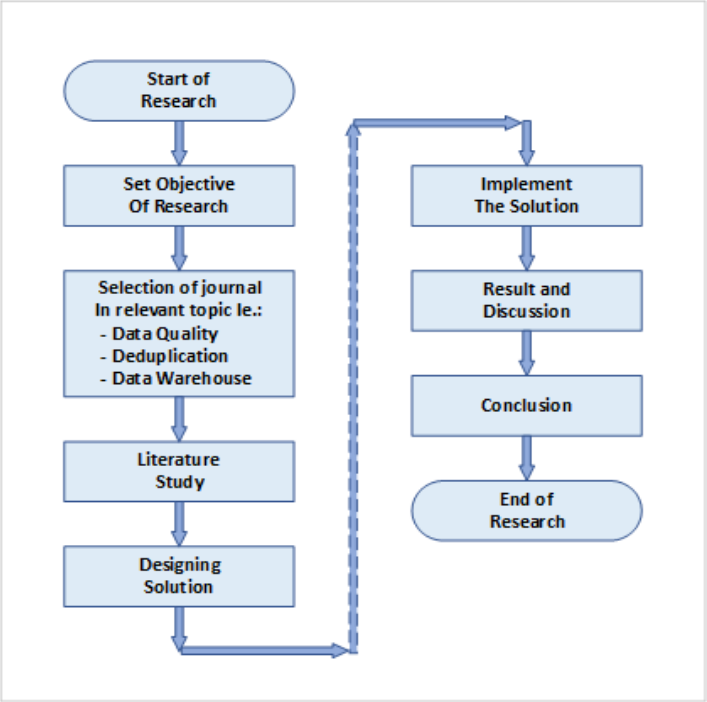


Gambar 2 *Surrogate Key Pipeline* (Kimball & Caserta, 2011)

Tabel 1 Berbagai Tipe SCD (Santos & Belo, 2011)

	Pendekatan per Atribut	Keuntungan	Kerugian
SCD Tipe-0	Jangan perbarui atribut	Tidak ada	Data yang kedaluwarsa
SCD Tipe-1	Perbarui atribut	Dimensi memiliki Data yang Diperbarui Kardinalitas dimensi tidak berubah.	Data Historis Hilang
SCD Tipe-2	Buat record baru setiap ada pembaruan, dan tandai record lama sebagai "Kedaluwarsa"	Data Historis disimpan	Peningkatan kardinalitas Dimensi Menangani Surrogate Keys
SCD Tipe-3	Simpan nilai sebelumnya dalam "Atribut lama" dan perbarui atribut	Kardinalitas dimensi tidak berubah	Hanya menyimpan nilai sebelumnya
SCD Tipe-4	Dimensi dibentuk oleh dua tabel. Satu berisi data saat ini dan lainnya berisi data historis	Tabel saat ini kecil Data Historis disimpan di tabel lain	Membutuhkan View untuk mengintegrasikan nilai dari dua tabel
SCD Tipe-6	Tipe-1 + Tipe-2 + Tipe-3 dikombinasikan	Data Historis disimpan Nilai sebelumnya mudah diakses	Peningkatan Dimensi kardinalitas Menangani Surrogate Keys

ISSN: 2654-3788
e-ISSN: 2654-4229



Gambar 3 Reasearch Methodology

3 Metode Penelitian

Dalam melakukan penelitian ini digunakan metodologi dengan tahapan yang ditunjukkan pada Gambar 3. Detail pelaksanaan tahapan tersebut Dijelaskan pada sub berikutnya.

3.1 Menentukan Tujuan Penelitian

Tujuan utama penelitian ini adalah untuk mencari cara dalam menangani *duplicate record* yang terjadi pada tabel Master karena tabel Master ini akan di gunakan sebagai *Dimensional Table* dalam *Data Warehouse*, Cara konvensional adalah menghapus atau menggabungkan Duplicated record tanpa mem-pertimbangkan apakah record tersebut mempunyai relasi ke tabel transaksi dari sistem operasional.

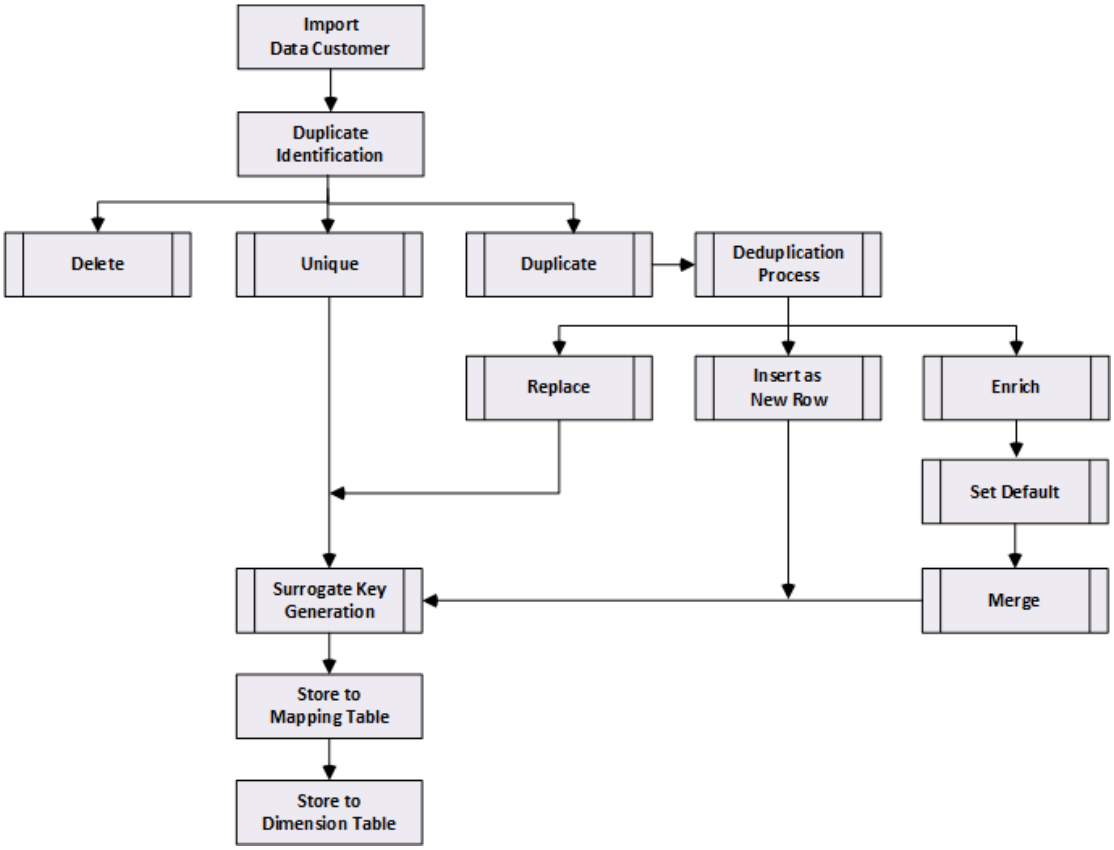
3.2 Mencari Jurnal yang Relefan

Mencari berbagai penelitian sebelumnya dalam perpustakaan digital baik dari http://e-resources.perpusnas.go.id/index.php maupun dari https://scholar.google.co.id/# yang berisi artikel penelitian paling relevan, dengan kata kunci "*Data Quality*", "*Data Duplication*", " *Data Warehouse*", dan "ETL".

3.3 Studi Pustaka

Mempelajari tentang kualitas data dan pengaruhnya pada bisnis, mempelajari tentang berbagai cara dalam menangani *duplicate record*, mempelajari mengenai mekanisme *Loading data* pada *Data Warehouse* termasuk mengeai *Slowly Changing Dimension* (SCD).

ISSN: 2654-3788
e-ISSN: 2654-4229



Gambar 4 Tahapan Solusi

3.4 Perancangan Solusi

Sistem ini bekerja secara bertahap seperti ditunjukkan pada Gambar 4, Langkah awal adalah menarik data Master yang akan dijadikan Dimensi yaitu pada proses Import Data Customer, proses berikutnya secara detail dijelaskan pada bagian 4. Hasil dan Diskusi.

3.5 Implementasi Solusi

Solusi dapat diimplementasikan dengan membangun aplikasi untuk mengidentifikasi dan menangani *Duplicate Record*s, lalu memodifikasi proses *Surogate Key Pipeline* menggunakan ***Pentaho Data Integration*** *(Meadows, Pulvirenti, & Roldá, 2013)*, hasil akhir akan terbentuk *ETL package* yang dapat digunakan untuk proses *Loading* dari sistem operasioanal ke *Data Warehouse*.

4 Hasil dan Diskusi

Evaluasi dari algorithma yang terbentuk dan hasilnya dijabarkan detailnya pada tulisan ini. Untuk dapat memahami permasalahan yang ada dapat diperhatikan data ***Customer*** pada Gambar 5 di bawah:

Customer

CUST.ID	FULL NAME	TEMPAT LAHIR
101	LEGI MAWAR LESTARI	JAKARTA
105	MEGA PRAHIKMAH	BOGOR
107	MIA MARYANI	BANTEN
123	MOH. DASLAN	JAKARTA
456	MOHAMMAD DASLAN	
300	NABILA AGESTIN	SOLO
312	NIA RAHMADANI	SURABAYA
423	NIA RAHMADHANI	SURABAYA
521	NUR ALFI LAIL	JAKARTA
223	NURJANAH	BANDUNG
321	OCTAVIANI AZIS ARNINGSIH	JAKARTA
434	OKTAVIANI	JAKARTA

Gambar 5 Tabel Customer

Pada tabel ***Customer*** tersebut tampak jika **Cust.ID=123** dan **Cust.ID=456** merujuk pada orang yang sama, namun pada kasus ini kita tidak bisa melakukan *merge* maupun *elimination* secara langsung dari kedua record tersebut karena kedua *record* tersebut telah mempunyai sejarah transaksi. Agar tabel *Customer* tersebut dapat digunakan pada *Data Warehouse* sebagai *Dimension Customer*, maka pada tabel tersebut tidak boleh ada *record* yang duplikat.

Dapat dilihat pada tabel *Transaction* pada Gambar 5 bahwa **Cust.ID=123** mempunyai

transasksi dengan **SO#=100** dengan nilai transaksi Rp. 125.000,- sedangkan **Cust.ID=456** mempunyai transasksi dengan **SO#=102** dengan nilai transaksi Rp. 450.000,-

Transaction

SO#	Date	Cust.ID	Prod.ID	QTY	Amount
100	13/08/2019	123	332	4	125.000
101	17/08/2019	200	632	2	34.000
102	21/08/2019	456	436	10	450.000
139	26/08/2019	226	332	6	120.000

Gambar 6 Tabel Transaksi

Process flow yang diajukan dalam menangani *duplicated record* pada sistem trasaksional semacam ini adalah sebagai berikut:

4.1 *Duplicate Identification*

Menurut (S. A. Babu, 2017) ada beberapa teknik yang telah diajukan dengan berbagai kerangka solusi untuk mengidentifikasi *duplicate record*. Teknik untuk menentukan bahwa dua record merujuk pada entitas yang sama dapat menjadi sangat komplek. Mengidentifikasi *duplicate record* adalah langkah yang vital dalam proses *data integration*. (Chandrasekar, 2013) Ada banyak algorithma untuk mengidentifikasi *duplicate record*. (Skandar et al., 2015).

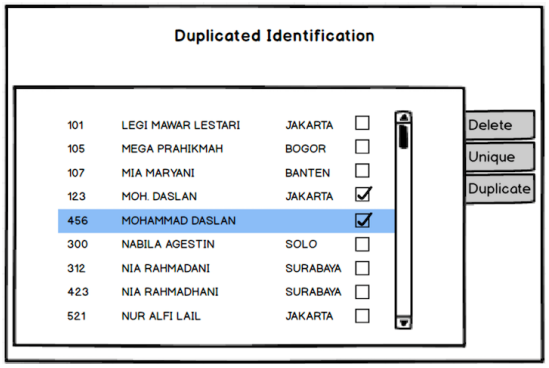
Masalah identifikasi intitas yang mirip dari realitas yang sama dalam dunia nyata melalui pencocokan yang tidak tepat pernah di teliti oleh (Tamilselvi & Gifta, 2011) yang mengidentifikasi data duplikat berdasarkan nilai ambang batas dan faktor kepastian dari sepasang record. Sementara (Sitas & Kapidakis, 2008) juga pernah mengembangkan algoritma untuk mengidentifikasi *duplicate record* menggunakan teknik *maching keys*. Untuk skala industry (Weis, Naumann, Jehle, Lufter, & Schuster, 2008) telah berhasil membuat prototype untuk melakukan identifikasi *duplicate record* pada *hierarchical* XML Data. Untuk situasi dengan waktu eksekusi yang terbatas dapat digunakan methode *progressive sorted neighborhood* dan methode *progressive blocking* yang dapat mengidentfikasi *duplikat record* dengan cepat dan efisien. (Papenbrock, Heise, & Naumann, 2014). Sampai saat ini identfikasi *duplicate record* masih menjadi topik yang sangat popular dalam penelitian.

Supervised Duplicate Identification adalah tehnik yang diajukan pada makalah ini, dengan membangun aplikasi untuk mempercepat dan mempermudah identifikasi dan penanganan *duplicate record*. Metode ini biasa digunakan untuk data dengan skala kecil-menengah. Aplikasi akan menampilkan data dari tabel ***Customer*** yang disortir berdasarkan *Primary Key* lalu data diidentifikasi secara visual seperti ditunjukkan pada Gambar 7.

Jika data yang masuk dikelompokkan sebagai sampah maka tekan tombol ***[Delete]***. Sebelum proses *Delete* sistem terlebih dahulu akan memeriksa apakah data tersebut terkait suatu transaksi apa tidak, jika tidak terkait suatu transaksi maka proses *Delete* bisa dilakukan, sebaliknya *Delete* tidak bisa dilakukan jika record tersebut ada pada tabel sejarah transaksi.

Beberapa *record* yang sudah terlihat unik (tidak ada duplikasi) seperti *record* dengan nama depan **Legi**, **Mega** dan **Mia** dapat dipilih dengan klik pada *option box* lalu klik tombol ***[Unique]***, kemudian dilanjutkan dengan proses pembuatan ***Surrogate Key*** atau Key penganti.

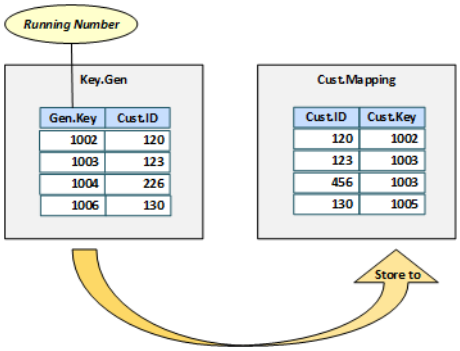
Untuk beberapa *record* yang diduga *Duplicate*, setelah dipilih lalu klik tombol ***[Duplicate]***, kemudian aplikasi akan masuk ke tahapan *Deduplication Process* seperti ditunjukkan pada Gambar 9.



Gambar 7 Duplicated Identification Dsiplay

4.2 *Surrogate Key Generation*

Surrogate Key dibentuk dengan cara meng-*insert* **Cust.ID** ke tabel **Key.Gen.** pada tabel ini field **Gen.Key** adalah ***running number field*** yang akan otomatis membentuk *sequence number* saat ada *record* baru yang disisipkan. ***running number*** yang terbentuk tersebut disimpan pada tabel **Cust.Mapping,** seperti ditunjukkan pada Gambar 8.



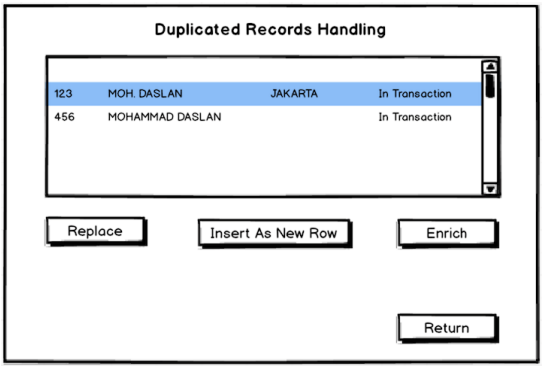
Gambar 8 Mapping Process

4.3 *Deduplication Process*

Menurut (Culotta & McCallum, 2005) *Record deduplication* adalah proses untuk mengabungkan (*merge*) beberapa *record* dari suatu database yang mengacu pada entitas yang sama. (D. R. Elkington, Zeng, & Morris, 2014) menjelaskan bahwa *machine learning* dapat digunakan dalam proses menggabungkan *duplicate record*.

Seperti telah disebutkan di atas bahwa dalam kasus ini kita tidak bisa melakukan *merge* maupun *eliminate* karena kedua *record* tersebut telah mempunyai sejarah transaksi, yang bisa diupayakan adalah *partial merge* yaitu *merge field-field* selain PK (*Primary Key*) nya, sedangkan PK dari masing-masing *duplicate record* harus tetap disimpan.

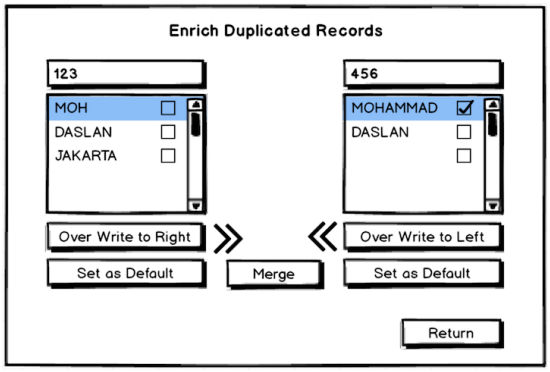
Desain aplikasi pada makalah ini menyediakan beberapa pilihan proses atas *Duplicated Records* seperti ditunjukkan pada Gambar 9 di bawah. Klik tombol ***[Replace]*** untuk menimpa record yang di bawah dengan record di sebelah atasnya, namun sebelum proses *Replace* dilakukan, sistem terlebih dahulu akan memeriksa apakah data tersebut terkait suatu transaksi apa tidak. Jika tidak terkait maka proses *Replace* akan dilakukan, sebaliknya *Replace* tidak bisa dilakukan jika record tersebut sudah ada pada tabel sejarah transaksi.



Gambar 9 Duplicated Records Handling Display

Jika kedua *record* akan diperlakukan sebagai entitas yang berbeda maka dapat di lakukan dengan cara klik tombol ***[Insert As New Row]***, namun jika kedua *duplicated record* akan diperlakukan sebagai entitas yang sama maka dapat dilakukan dengan cara klik tombol ***[Enrich]***, detail proses *Enrich* ditunjukkan seperti pada Gambar 10 di bawah.

Pada contoh kasus ini **Cust.ID=123** ada disebelah kiri dan **Cust.ID=456** sebelah kanan, pada *field* **Cust.FName** akan diambil value dari **Cust.ID=456** yaitu "Mohammad" sedangkan untuk field Cust.Address akan diambil value dari **Cust.ID=123** yaitu "Jakarta". Dengan aplikasi akan dipermudah untuk memilih data dari record yang mana yang akan digunakan.



Gambar 10 Merging Duplicated Records Display

Setelah itu pilih record yang akan digunakan datanya degan klik ***[Set as Default]*** pada sebelah kiri atau kanan, dalam kasus ini klik ***[Set as Default]*** pada sebelah kiri lalu klik tombol ***[Merge]***. Tombol ***[Merge]*** akan *disable* sebelum tombol ***[Set as Default]*** diklik.

Proses berikutnya adalah proses untuk membentuk ***Common Surrogate Key*** untuk kedua record yang telah *merged*. **Cust.ID** yang dipilih sebagai *Default* yaitu **Cust.ID=123** disisipkan ke

tabel **Key.Gen**, seperti sudah dijelaskan di atas, pada tabel ini field **Gen.Key** adalah ***running number*** yang akan otomatis men-*generate sequence number* saat ada *record* baru yang disisipkan. ***running number*** yang terbentuk adalah **Gen.Key = 1003**. **Common Key** dari kedua record tersebut beserta kedua **Cust.ID** dari kedua *Duplicated Record* disimpan pada tabel **Cust.Mapping,** seperti ditunjukkan pada Gambar 8 di atas. Hasil akhir pada tabel ***Dim_Customer*** seperti ditunjukkan pada Gambar 11 di bawah.

Dim_Customer

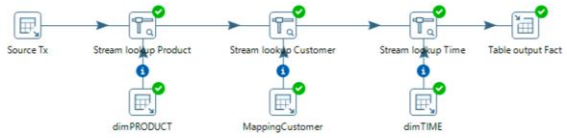
Cust.Key	Cust.Name	Cust.Address
1003	Mohammad Daslan	Jakarta
1004	Ferdian Nohan	Surabaya
1005	Donna Rawatih	Yogya
1323	Yunita Budiman	Bandung

Gambar 11 Tabel Dim_Customer

4.4 *Modified Surrogate Key Pipeline*

Untuk Loading data transasksi ke Data Warehouse diperlukan proses Surrogate Key Pipeline (Kimball & Caserta, 2011), proses ini untuk memasukkan data transaksi dari sistem operasional ke tabel Fact pada database Data Warehouse. Khusus untuk dimensi Customer proses ini perlu dimodifikasi, proses **lookup** berdasarkan **Cust.ID** yang awalnya ke tabel **dim_Customer** perlu dialihkan ke tabel **Cust.Mapping** untuk mendapatkan **Cust.Key** sebagai Surrogate Key Customer. **Cust.Key** ini nantinya yang akan disimpan pada tabel **Fact** sebagai **Foreign Key (FK)**.

Proses Modified Surrogate Key Pipeline ini dapat diemplementasikan pada **Pentaho Data Integration** sebagai Transformation package Gambar 12 berikut:



Gambar 12 Pentaho ETL Package

Setelah data transaksional dari sistem operasional di Load ke Data Warehouse, maka akan didapatkan tabel **Fact** dan **Dim_Customer** seperti yang ditunjukkan pada Gambar 13 di bawah. Dapat dilihat untuk Customer **Muhammad Daslan** dengan **Cust.Key=1003** mempunyai dua transaksi yaitu **SO#=100** dan **SO#=102**.

Fact

SO#	Date	Stor.Key	Cust.Key	Prod.Key	QTY	Amount
100	13/08/2019	5980	1003	2012	4	125.000
101	17/08/2019	5400	1005	2331	2	34.000
102	21/08/2019	5634	1003	2433	10	450.000
409	25/09/2019	5400	1023	2012	16	920.000

Dim_Customer

Cust.Key	Cust.Name	Cust.Address
1003	Mohammad Daslan	Jakarta
1004	Ferdian Nohan	Surabaya
1005	Donna Rawatih	Yogya
1323	Yunita Budiman	Bandung

Gambar 13 Fact-Dimension Relation

5 Simpulan

Data Kotor khususnya *Duplicated Record* adalah hal yang tidak bisa dihindarkan dalam berbagai data transaksi pada banyak perusahan. Berbagai Teknik dan algoritma telah diajukan dalam beberapa penelitian untuk mengidentifikasi dan menangani *Duplicated Record*.

Sistem pelaporan dan Analisa membutuhkan data yang berkualitas agar informasi yang dihasilkan dapat meningkatkan proses pengambilan keputusan, termasuk di sini adalah *Business Intelligence* dan *Data Warehouse* sebagai komponen utamanya.

Metode untuk mengidentifikasi *duplicated record* sangat beragam dan bisa menjadi sangat komplek, namun dengan membangun aplikasi kecil kerumitan tersebut dapat diuraikan. Jika data dari sistem transaksional khususnya data master akan di-*load* ke *Data Warehouse* sebagai *Dimensional Table* maka perlu perlakuan khusus dalam proses *deduplication*, karena data master tersebut boleh jadi sudah terkait pada sejarah transaksi sehingga tidak bisa di eliminasi atau di *merge*. strategi untuk menyelesaikan masalah ini adalah dengan menambahkan tabel *mapping* dan *rutine* yang mengkonversikan *Natural ID* menjadi *Surrogate Key* pada proses *Modified Surrogate Key Pipeline.*

6 Recommendations

Untuk meningkatkan kemampuan *Supervised Duplicate Identification* dengan jumlah *master data* yang lebih besar maka dibutuhkan filter yang canggih agar data yang memerlukan supervisi manual seminimal mungkin.

Dengan demikian maka sebagian besar *records* yang sudah pasti bersifat *Unique* dapat secara otomatis diproses oleh aplikasi. Algorithma untuk mengidentifikasi kemiripan pada tingkat tertentu suatu records akan dapat mempercepat proses pada aplikasi ini.

Referensi

Babu, K. (2012). Business intelligence: Concepts, components, techniques and benefits. *Components, Techniques and Benefits (September 22, 2012)*.

Babu, S. A. (2017). Duplicate Record Detection and Replacement within a Relational Database. *Advances in Computational Sciences and Technology, 10*(6), 1893-1901.

Bajpai, J., & Metkewar, P. S. (2016). Data quality issues and current approaches to data cleaning process in data warehousing. *Glob. Res. Dev. J. Eng, 1*(10), 14-18.

Chandrasekar, C. (2013). An optimized approach of modified bat algorithm to record deduplication. *International Journal of Computer Applications, 62*(1).

Culotta, A., & McCallum, A. (2005). *Joint deduplication of multiple record types in relational data*. Paper presented at the Proceedings of the 14th ACM international conference on Information and knowledge management.

Elkington, D., Zeng, X., & Morris, R. (2016). Resolving and merging duplicate records using machine learning. In: Google Patents.

Elkington, D. R., Zeng, X., & Morris, R. G. (2014). Resolving and merging duplicate records using machine learning. In: Google Patents.

Fleckenstein, M., & Fellows, L. (2018). Data Warehousing and Business Intelligence. In *Modern Data Strategy* (pp. 121-131): Springer.

Haug, A., Zachariassen, F., & Van Liempd, D. (2011). The costs of poor data quality. *Journal of Industrial Engineering and Management (JIEM), 4*(2), 168-193.

Ker, C., VAISHNAV, P., & Dvinov, D. (2017). Merging multiple groups of records containing duplicates. In: Google Patents.

Kimball, R., & Caserta, J. (2011). *The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data*: John Wiley & Sons.

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*: John Wiley & Sons.

Marsh, R. (2005). Drowning in dirty data? It's time to sink or swim: A four-stage methodology for total data quality management. *Journal of Database Marketing & Customer Strategy Management, 12*(2), 105-112.

Meadows, A., Pulvirenti, A. n. S., & Roldá, M. a. C. (2013). *Pentaho Data Integration Cookbook : Over 100 Recipes for Building Open Source ETL Solutions with Pentaho Data Integration* (Vol. Second edition). Birmingham: Packt Publishing.

Papenbrock, T., Heise, A., & Naumann, F. (2014). Progressive duplicate detection. *IEEE Transactions on knowledge and data engineering, 27*(5), 1316-1329.

Santos, V., & Belo, O. (2011). *No need to type slowly changing dimensions.* Paper presented at the IADIS International Conference Information Systems.

Sitas, A., & Kapidakis, S. (2008). Duplicate detection algorithms of bibliographic descriptions. *Library Hi Tech, 26*(2), 287-301.

Skandar, A., Rehman, M., & Anjum, M. (2015). An Efficient Duplication Record Detection Algorithm for Data Cleansing. *International Journal of Computer Applications, 127*(6), 28-37.

Tamilselvi, J. J., & Gifta, C. B. (2011). Handling duplicate data in data warehouse for data mining. *International Journal of Computer Applications, 15*(4), 7-15.

Tamilselvi, J. J., & Saravanan, V. (2009). Detection and elimination of duplicate data using token-based method for a data warehouse: A clustering based approach. *International Journal of Dynamics of Fluids, 5*(2), 145-164.

Weis, M., Naumann, F., Jehle, U., Lufter, J., & Schuster, H. (2008). Industry-scale duplicate detection. *Proceedings of the VLDB Endowment, 1*(2), 1253-1264.