PROPOSAL PENELITIAN

PEMBANGKITAN SINGLE CUSTOMER INFORMATION FILES (CIF) MENGGUNAKAN MOD-EIM: STUDI KASUS BANK DANAMON INDONESIA

Oleh

HENDRIK - G651160051

DEPARTEMEN ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

INSTITUT PERTANIAN BOGOR

2017

PEMBANGKITAN SINGLE CUSTOMER INFORMATION FILES (CIF) MENGGUNAKAN MOD-EIM: STUDI KASUS BANK DANAMON INDONESIA

Hendrik[1], Agus Buono[2], Yani Nurhadryani[3]
[1]Mahasiswa Departemen Ilmu Komputer, FMIPA IPB
[2]Ketua Komisi Pembimbing, staf pengajar Departemen Ilmu Komputer, FMIPA IPB
[3]Anggota Komisi Pembimbing, staf pengajar Departemen Ilmu Komputer, FMIPA IPB

Abstract

Information management in an enterprise typically involves integrating data from across the enterprise and beyond, cleansing the data, matching the data to remove any duplicates, standardizing the data, and then storing the data in a centralized location in order to ensure the data quality. Cleansing and matching data from various source is the hardest challenge to deal with it. The data usually have duplication and hard to decide which is the best data is. Bank Danamon Indonesia (BDI) as enterprise in banking has requirement to develop Information management in customer information files (CIF) data. It's called as E-CIF system. E-CIF is developed to ensure the BDI's costumer information data quality and generate single valid CIF called single CIF. Single CIF is used for the next business strategy and to support the business process runs effective and efficient. E-CIF uses Modified Enterprise Information Management (Mod-EIM) which consist of several modified processes from its predecessor.

Keyword: *managing information, costumer information file (CIF), mod-EIM*

PENDAHULUAN

Latar Belakang

Customer relationship management (CRM) tengah dikembangkan oleh Bank Danamon Indonesia (BDI). CRM dikembangkan agar BDI dapat menjaga hubungan dengan nasabah melalui pemahaman terhadap latar belakang nasabah dan perilakunya, meningkatkan layanan kepada nasabah, dan mengembangkan inovasi produk melalui implementasi layanan interaktif dan peningkatan teknologi pada basis data nasabah (Verhoef, 2001). CRM membutuhkan data nasabah yang valid untuk memberikan hasil analisis yang akurat dan efektif sehingga memberikan layanan yang optimal untuk nasabah (Gupta, 2012). Istilah satu data nasabah valid dikenal dengan *single customer view*. *Single coustmer view* adalah representasi agregat, konsisten dan holistik dari data pelanggan yang dapat diakses di satu tempat (Riverleen, 2011).

BDI menyimpan informasi mengenai data nasabah pada *customer information files* (CIF). CIF merupakan sebuah fail, baik elektronik atau fisik, yang menyimpan semua informasi yang bersangkutan mengenai informasi pribadi dan rekening nasabah (Beall, 2003). CIF pada kasus kali ini merupakan fail elektronik yang tersimpan pada basis data. CIF pada BDI terdiri dari CIF nasabah individu dan CIF nasabah perusahaan. Sebagai contoh, CIF individu memuat informasi mengenai nama, alamat, tanggal lahir, tempat lahir dan ibu kandung.

Dalam pengembangan CRM, BDI menghadapi beberapa tantangan antara lain, CIF yang terpisah di tiga sumber data, yaitu pada New Core Business System (NCBS), Credit card (Ascend), dan Adira Finance. Selain itu, ukuran basis data yang besar atau *Big Data*, yaitu mencapai 200 GB. Kemudian adanya duplikasi CIF di antara tiga sumber data serta CIF yang mengandung nilai *fuzzy* menyebabkan BDI sulit untuk melakukan identifikasi *single* CIF. Duplikasi data mencapai 30% dari keseluruhan data (BDI, 2012). Duplikasi data perlu diperbaiki, karena akan mempengaruhi hasil ekstraksi informasi, mengacaukan hasil analisis,

dan membuat peluang kesalahan pada penentuan keputusan lebih tinggi (Guo, 2012).

Enterprise information management (EIM) diinisiasi oleh Michael R Thompson (2003) merupakan solusi untuk memastikan kualitas data atau informasi pada level *enterprise*. EIM adalah cabang disiplin pada bisnis strategis yang menggabungkan banyak prinsip utama integrasi perusahaan, *business intelligence* (BI), dan manajemen konten untuk merampingkan dan memformalkan aktivitas yang terkait dengan penyimpanan data, akses, dan penanganan. Inisiatif EIM yang komprehensif memadukan proses dan teknologi untuk memperbaiki secara signifikan cara pengelolaan dan pengelolaan informasi di seluruh perusahaan.

Dengan EIM, organisasi dapat meningkatkan nilai informasi perusahaan mereka, memanfaatkannya untuk meningkatkan produktivitas operasional, mengurangi biaya *overhead*, dan memperoleh keunggulan kompetitif yang substansial. EIM secara umum meliputi tahapan integrasi berbagai sumber data, pembersihan data, pencocokan data untuk menghapus duplikasi, membuat standar model data, dan menyimpan data pada satu lokasi pusat gudang data (Lam, 2009). Pembersihan dan pencocokan data dari tahap integrasi adalah tahapan dan tantangan tersulit dari EIM (Kolb, 2009).

Data hasil integrasi biasanya memiliki duplikasi, memiliki nilai *fuzzy* dan cukup sulit menentukan mana data terbaik. Penelitian mengenai pencocokan data yang memiliki nilai *fuzzy* dan duplikasi telah banyak dilakukan. Teknik pencocokan yang dikembangkan banyak mengadopsi atau peningkatan dari algoritma *fuzzy clustering*. Salah satu algoritma *fuzzy clutering* yang banyak digunakan adalah algoritma Fuzzy C-Means (FCM). Algoritma FCM dapat membagi sebuah himpunan terbatas dari n elemen ke dalam sebuah himpunan dari c kelompok *fuzzy* dengan memberikan beberapa kriteria (Jian, 2007).

Guo *et al* (2012) mengadopsi teknik *fuzzy c-means clustering* dikombinasikan dengan jarak Levenshtein untuk melakukan pembersihan dan pencocokan duplikasi data. Teknik tersebut secara akurat dan cepat dapat mendeteksi dan menghilangkan duplikasi data. *Recall* dan *precision* deteksi duplikasi data meningkat sebanyak 50%. Andrejková *et al* (2013), menerapkan pendekatan Fuzzy Logic pada fungsi *pattern matching* untuk menentukan *finite state automaton* atau batasan keadaan kapan karakter dikatakan mirip. Hasilnya *pattern matching* hanya membangkitkan karakter yang cocok dan tidak menghasilkan *error*.

Selain itu, untuk menangani Big Data, Prabha et all (2014) berhasil melakukan reduksi data menggunakan teknik peningkatan dari Fuzzy Clustering, yaitu *Incremental Weighted Fuzzy C-Means* (IWFCM). IWFCM memasukkan bobot sebagai parameter yang menggambarkan pengaruh masing-masing obyek di dalam *cluster*. IWFCM diterapkan pada *e-book dataset*. Dataset tersebut diolah pada lingkungan Hadoop dengan *map reduce framework*. IWFCM berhasil menghasilkan *cluster* dengan minimum *run time* dan kualitas yang baik.

Enterprise Customer Information Files (E-CIF) dikembangkan oleh BDI untuk menangani permasalahan duplikasi pada data nasabah hasil penyatuan dari berbagai sumber data. E-CIF merupakan sistem yang mengombinasikan data nasabah di berbagai sumber data. E-CIF bertujuan untuk menciptakan satu data nasabah yang valid dan mencakup keseluruhan informasi dan relasi yang terkait dengan nasabah. Saat ini, BDI belum memiliki *single* CIF.

E-CIF sendiri merupakan pengembangan dari aplikasi Master Customer Information Files (M- CIF) yang saat ini sudah tersedia di Bank Danamon Indonesia. M-CIF melakukan pembangkitan *single* CIF menggunakan EIM yang dikembangkan oleh Microsoft pada produknya, yaitu Microsoft SQL Server 2014. Namun, metode *matching* atau pencocokan pada

EIM yang telah digunakan masih belum sesuai dengan kebutuhan BDI. CIF yang dihasilkan tidak sesuai dengan kebutuhan BDI, yaitu masih memiliki banyak duplikasi CIF.

BDI membutuhkan skenario pembangkitan *single* CIF baru pada E-CIF. Skenario baru ini dapat melakukan integrasi data, pembersihan data, pengelompokan data, pencocokan data, verifikasi data dan koreksi data. Skenario pembangkitan tersebut nantinya juga dapat menangani data yang memiliki nilai *fuzzy*. Data yang berkualitas yang diperlukan adalah data *single* CIF yang tidak memiliki duplikasi serta isi dari setiap data merupakan isian yang *valid*.

EIM pada SQL Server 2014 memiliki tiga komponen *services* utama, yaitu *Mater Data Service* (MDS), *Integraton Service* (IS) dan *Data Quality Services* (DQS). EIM SQL Server 2014 memberikan beberapa opsi operasi untuk melakukan pembangkitan dan pengelolaan data. Operasi untuk melakukan manajemen data berbasis pengetahuan, pembersihan data, dan pencocokan data dapat dilakukan pada *data quality service* (DQS). Selain itu, pada komponen *integration service* (IS) atau lebih dikenal dengam SSIS, dapat dilakukan operasi pembersihan data menggunakan *fuzzy lookup*, dan *lookup fuzzy transformation* dan operasi tersebut dapat dilakukan secara *batch* tanpa pengawasan. Sedangkan, *management data service* (MDS) memiliki kemampuan untuk melakukan operasi pembuatan model dan basis pengetahuan.

Tantangan yang cukup signifikan pada penelitian ini adalah mengembangkan sebuah skenario EIM baru agar operasi algoritma *fuzzy matching* dapat berjalan secara akurat, efektif dan efisien untuk menghasilkan CIF. Selain itu, penerapan pada lingkungan *open source* agar tercapai efisiensi biaya, karena permasalahan lisensi produk Microsoft yang cukup mahal.

Pada penelitian ini, skenario EIM pada Microsoft SQL Server 2014 ini akan dimodifikasi dan dikembangkan pada lingkungan *open source* Apache Spark dan Apache Hadoop. Apache Spark dikenal merupakan *framework* dengan performa *runtime* yang baik untuk melakukan *clustering* pada Big Data. Sedangkan Hadoop sendiri dikenal sebagai *framerowk* penyimpanan Big Data yang memiliki performa mumpuni.

Manfaat Penelitian

ECI-F dapat mengidentifikasi dan mengombinasikan data maupun relasi nasabah sehingga menyediakan *single CIF* untuk aplikasi pada CRM atau pihak lain yang terkait (contoh: Manulife). Dengan demikian, CRM dapat memberikan analisis yang akurat sehingga membantu pertumbuhan bisnis Bank Danamon Indonesia secara signifikan.

Tujuan Penelitian

Tujuan dari penelitian adalah membuat sebuah skenario penyatuan, pembersihan, pengelompokan, pencocokan, dan evaluasi data nasabah atau CIF Bank Danamon Indonesia yang kemudian disebut *Modified Enterprise Information Management (mod-EIM)* untuk menghasilkan *single* CIF Bank Danamon Indonesia.

Ruang Lingkup Penelitian

E-CIF dikembangkan dengan menggunakan data nasabah Bank Danamon Indonesia dan digunakan untuk Bank Danamon Indonesia. E-CIF dibangun pada lingkungan teknologi Microsoft .NET, SQL Server Enterprise 2014, Apache Spark dan Apache Hadoop. Sumber data nasabah berasal dari tiga sumber data, yaitu Core Banking, Ascend, dan Adira. Data yang digunakan adalah data nasabah individual.

Tinjauan Pustaka

Enterprise Information Management (EIM)

Enterprise information management (EIM) diinisiasi oleh Michael R Thompson (2003) merupakan solusi untuk memastikan kualitas data atau informasi pada level *enterprise*. EIM adalah cabang disiplin pada bisnis strategis yang menggabungkan banyak prinsip utama integrasi perusahaan, *business intelligence* (BI), dan manajemen konten untuk merampingkan dan memformalkan aktivitas yang terkait dengan penyimpanan data, akses, dan penanganan. Inisiatif EIM yang komprehensif memadukan proses dan teknologi untuk memperbaiki secara signifikan cara pengelolaan dan pengelolaan informasi di seluruh perusahaan.

Dengan EIM, organisasi dapat meningkatkan nilai informasi perusahaan mereka, memanfaatkannya untuk meningkatkan produktivitas operasional, mengurangi biaya *overhead*, dan memperoleh keunggulan kompetitif yang substansial. EIM secara umum meliputi tahapan integrasi berbagai sumber data, pembersihan data, pencocokan data untuk menghapus duplikasi, membuat standar model data, dan menyimpan data pada satu lokasi pusat gudang data (Lam, 2009). Pembersihan dan pencocokan data dari tahap integrasi adalah tahapan dan tantangan tersulit dari EIM (Kolb, 2009).

Big Data

"Big Data" adalah sebuah istilah untuk merepresentasikan sebuah koleksi data yang berukuran besar, berstruktur kompleks dan mengalir dengan kecepatan tinggi yang mana tidak bisa dikelola menggunakan teknologi *traditional data-warehouse*. Big Data adalah dunia data yang berada di luar tradisional *data warehouse* dan *entreprise*. Data tersebut dihasilkan oleh aktivitas perangkat, berita pada blog dan sosial media, data sensor, dan transaksi perdagangan. Big Data berbentuk tidak terstuktur, tidak terfilter dan berbentuk *non-relational*.

Menurut Gartner, "Big Data adalah aset data yang memiliki volume tinggi, kecepatan tinggi, dan keragaman tinggi yang membutuhkan metode pemrosesan baru unutk memungkinkan pembuatan keputusan, penemuan pengetahuan, dan optimasi proses. Selain itu, menurut Steve Stood, Big data adalah ketika aplikasi normal dari teknologi yang sedang digunakan tidak mampu membuat pengguna mendapatkan kebutuhan pengetahuan dari data yang dengan cepat dan biaya yang murah.

Apache Hadoop

Apache Hadoop adalah kerangka perangkat lunak *open-source* yang digunakan untuk penyimpanan terdistribusi dan pemrosesan *dataset* data besar dengan menggunakan model pemrograman MapReduce. Ini terdiri dari kumpulan komputer yang dibangun dari perangkat keras. Inti dari Apache Hadoop terdiri dari bagian penyimpanan, yang dikenal sebagai Hadoop Distributed File System (HDFS), dan bagian pemrosesan yang merupakan model pemrograman MapReduce. Hadoop membagi file menjadi blok besar dan mendistribusikannya ke node dalam sebuah cluster. Kemudian transfer kode paket ke dalam node untuk memproses data secara paralel. Pendekatan ini memanfaatkan wilayah data, [3] dimana node memanipulasi data yang mereka akses.

Kerangka Apache Hadoop terdiri dari empat modul antara lain: 1) Hadoop Common - berisi perpustakaan dan utilitas yang dibutuhkan oleh modul Hadoop lainnya, 2) Hadoop Distributed File System (HDFS) - sistem file terdistribusi yang menyimpan data pada mesin,

memberikan *bandwidth* agregat yang sangat tinggi ke seluruh *cluster*, 3) Hadoop YARN - platform yang bertanggung jawab untuk mengelola sumber daya komputasi dan menggunakannya untuk melakukan penjadwalan aplikasi, dan 4) Hadoop MapReduce - sebuah implementasi dari model pemrograman MapReduce untuk pemrosesan data berskala besar.

Fuzzy Matching

Fuzzy matching adalah sebuah mekanisme pencocokan yang dibantu oleh komputer, untuk mencocokkan kata, frasa, kalimat atau sebagin bagian teks dari kalimat pada sebuah basis data. Algoritma pencocokan fuzzy digunakan untuk membandingkan dua *string* dengan mengukur jumlah karakter yang harus dimodifikasi (menambahkan, menghapus atau mengubah) *source string* agar terlihat seperti *target string*. Sebagai contoh, jika *source string* adalah '*tour'* dan *target string* adalah '*tow'*, algoritma akan mengembalikan *score* 2, karena melakukan dua penggantian karakter.

Levenshtein Distance

Secara matematis, perhitungan jarak **Levenshtein** antara dua *string* a,b (dari masing-masing panjang | a | dan | b |) diberikan oleh formula $\mathbf{Lev_{a,b}}$ (| a |, | b |),

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

dimana $1_{(ai\ !=\ bj)}$ adalah fungsi indikator. Jarak akan bernilai 0 ketika $a_i = b_j$ dan jika keadaan lain akan bernilai 1. $\mathbf{Lev_{a,b}}$ (| a |, | b |) adalah jarak antara karakter pertama *i* dari *a*, dan karakter pertama *j* dari *b*.

Misalnya, jarak Levenshtein antara "kitten" dan "sitting" adalah 3, karena tiga pergantian karakter berikut mengubah satu ke yang lainnya, dan tidak ada cara lain untuk melakukannya sebanyak kurang dari tiga pergantian karakter tersebut:

Kitten → sitten (substitusi dari "s" untuk "k")
Sitten → sittin (pengganti "i" untuk "e")
Sittin → duduk (penyisipan "g" di bagian akhir).

Damerau-Levenshtein Distance

Jarak Damerau-Levenshtein (dinamai menurut Frederick J. Damerau dan Vladimir I. Levenshtein) adalah metrik *string* untuk mengukur jarak antara dua urutan. Secara informal, jarak Damerau-Levenshtein antara dua kata adalah jumlah operasi minimum (terdiri dari penyisipan, penghapusan atau penggantian karakter tunggal, atau transposisi dua karakter yang berdekatan) yang diperlukan untuk mengubah satu kata ke kata lainnya.

Dalam makalah manuskripnya, Damerau menyatakan bahwa keempat operasi ini sangat sesuai untuk menangani lebih dari 80% dari semua kesalahan ejaan manusia. Sementara motivasi awalnya adalah mengukur jarak antara kesalahan ejaan manusia dan memperbaiki ejaan.

Secara matematis jarak Damerau–Levenshtein di antara dua strings *a* dan *b* adalah sebuah fungsi $\boldsymbol{d_{a,b}\ (i,j)}$, yang nilainya terletak antara sebuah *i*–symbol prefix (inisial *substring*) dari

string a dan *a j–symbol prefix* dari *b*.

Fungsi $d_{a,b}(i,j)$ didefiniskan secara rekursif seperti formula di bawah ini:

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \\ d_{a,b}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

dimana 1 (a_i != b_j) adalah fungsi indikator, sebanding dengan 0 ketika $a_i = b_j$ dan kondisi lainnya bernilai 0.

Fuzzy c-Means Clustering

Fuzzy C-means (FCM) clustering memungkinkan sebuah grup data masuk ke dalam dua atau lebih dari dua grup. Metode ini diusulkan oleh Dunn pada tahun 1973, dan dikembangkan oleh Bezdek pada tahun 1981. Metode ini merupakan pengelompokan didasarkan pada pembagian dan bertujuan agar obyek yang mirip masuk ke dalam kelompok yang sama menjadi maksimum, dan obyek yang berbeda masuk ke kelompok sama menjadi minimum. Hal ini bergantung pada fungsi obyektif yang meminimalkan, dapat dilihat pada gambar di bawah ini:

$$Jm = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|_2 \quad (1)$$

di mana m adalah setiap bilangan *real* lebih besar dari 1, m diberikan nilai konstanta 2.00. u_{ij} adalah tingkat j dari anggota grup x_i. x adalah pengukuran dimensi i; C adalah *ij* dimensi kluster, * pusat adalah untuk menggambarkan kesamaan antara setiap kriteria, setiap pusat pengukuran. Fungsi tujuan dioptimalkan melalui daftar fungsi berulang seperti persamaan (1), kemudian untuk memperbarui anggota *uij* menggunakan persamaan (2) dan pusat grup C_j diperbaharui menggunakan persamaan (3):

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \quad (2)$$

$$C = \frac{\sum_{i=1}^{N} u_{ij}^{m} \bullet x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Iterasi berhenti ketika

$$max_{ij} = \left\{ \left| u_{ij}^{(k-1)} - u_{ij}^{k} \right| \right\} \varepsilon \quad (3)$$

Alat dan Data

Pada penelitian ini menggunakan alat berupa piranti keras dan piranti lunak. Piranti keras yang digunakan, yaitu Macbook Pro Retina 2015 untuk piranti *developement* dan Server HP DL360 untuk piranti *deployment*. Piranti lunak terdiri atas *integrated development environment* (IDE) IntelliJ IDEA dan Apache Spark. IDE tersebut digunakan untuk melakukan pengolahan data atau *matching process* dengan pemrograman bahasa Scala.

Selain itu, IDE Microsoft Visual Studio 2013 digunakan untuk mengembangkan *end-user website* E-CIF. Bahasa pemrograman menggunakan bahasa C# dan *framework* ASP Net MVC dan DevExpress. Sedangkan penyimpanan data pada tahap pengolahan data dipusatkan di Hadoop HDFS dan penyimpanan data pada tahap final dipusatkan di SQL Server 2014.

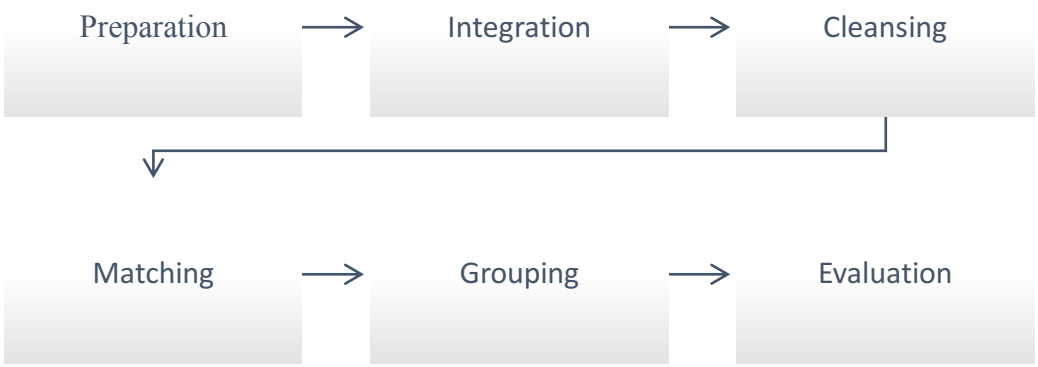
Data

Data yang digunakan pada penelitian ini adalah data nasabah Bank Danamon Indonesia. Data nasabah didapatkan dari tiga basis data di Bank Danamon Indonesia. Tiga basis data tersebut adalah Ascend, NCBS, dan Adira. Sedangkan nasabah yang digunakan adalah nasabah tipe individual/personal. Struktur dan sampel data nasabah personal yang ada pada salah satu basis data dapat dilihat pada Gambar 2. Pada Gambar 2 dapat dilihat beberapa baris duplikasi data. Dapat dilihat pada kolom *address, city, Firstname, LastName*, dan *zip* memiliki nilai/isian yang mirip.

Gambar 2. Struktur dan sampel data nasabah individual

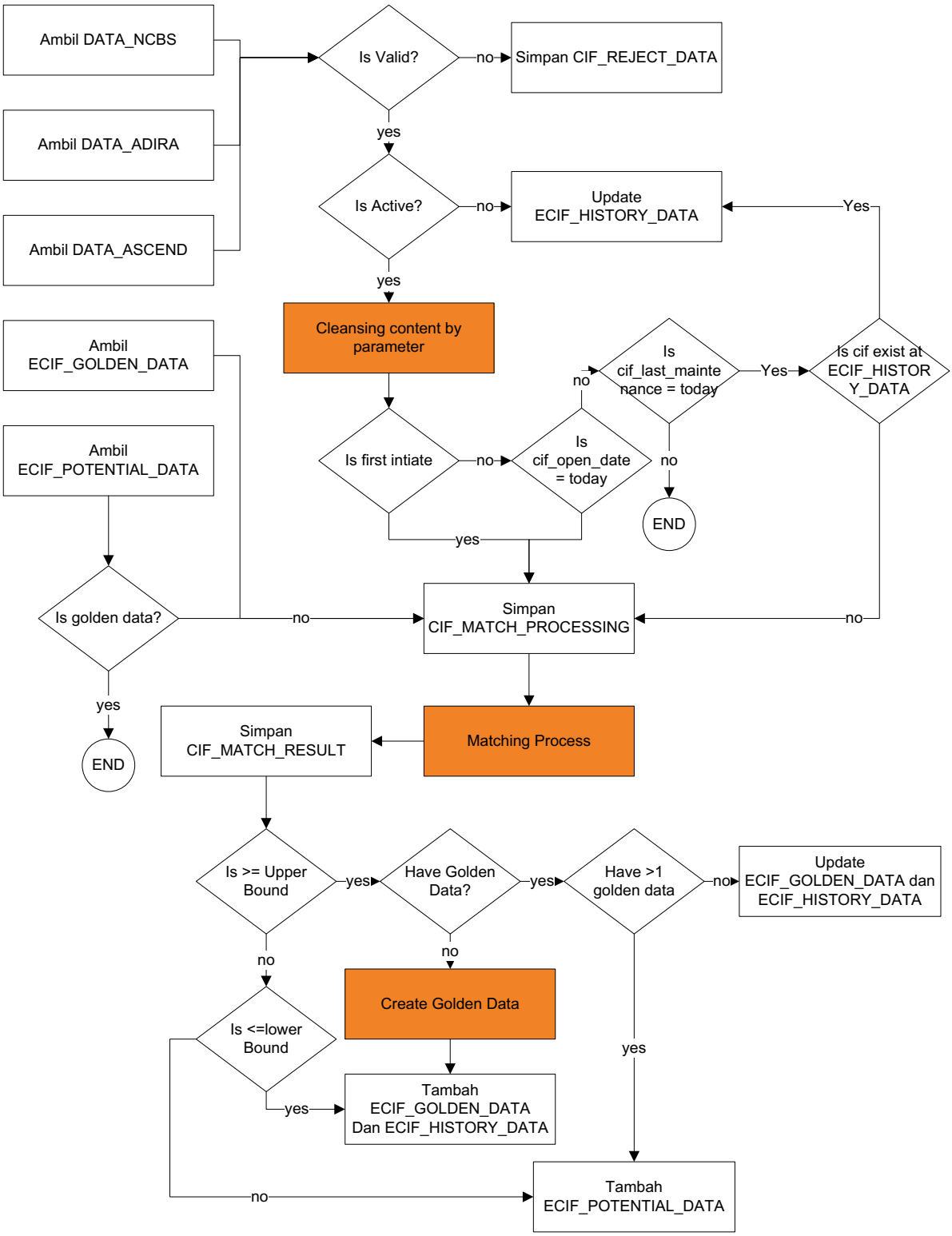
Metode

Metode untuk menghasilkan *single* CIF meliputi tahapan penyatuan, pemodelan, pembersihan, pencocokan, dan pengelompokan data disebut *modified enterprise information management (Mod-EIM)*. Skema *general* skenario Mod-EIM dapat dilihat pada Gambar 2.



Gambar 3. Skenario Mod-EIM

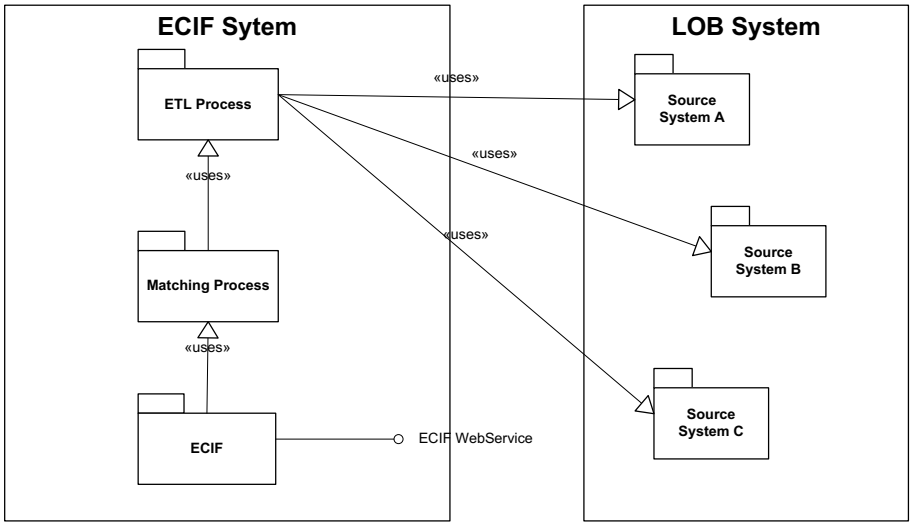
Skema lebih lengkap Mod-EIM dapat dilihat pada pada Gambar 3. Pada skema tersebut digambarkan rangkain utuh pembangkitan *single* CIF yang disebut "Generating Single CIF". Berikut ini adalah skema yang menggambarkan tiga rangkaian sub proses, yakni proses Cleansing, Matching/Fuzzy Grouping, dan Generate/Create Golden Data.



Gambar 2. Alur *generating single CIF*

1. Preparation and Integration

Mod-EIM dimulai dengan persiapan dan integrasi. Persiapan dalam hal ini adalah pembuatan arsitektur yang dibagi menjadi dua entitas besar, yaitu sistem ECIF dan sistem LOB. Sistem ECIF terdiri atas pengolahan data oleh *engine*, evaluasi CIF pada *end-user* ECIF dan proses ETL. Sedangkan sistem LOB berisi integrasi tiga basis data nasabah yaitu Ascend (sumber A), Adira (sumber B) dan NCBS (sumber C). Sistem LOB menggunakan *framewok* Hadoop untuk menyimpan data berukuran besar pada HDFS. Hadoop memindahkan data berukuran besar dari berbagai sumber ke tempat sentral *data warehouse* SQL Server 2014. ETL akan mengonsumsi data dari ketiga sumber data melalui sumber data pada Hadoop. Sebagian besar operasi ETL adalah melakukan pemetaan dan sinkronisasi data nasabah agar menjadi satu sumber data nasabah yang terstandarisasi. Skenario integrasi dapat dilihat pada Gambar 3.



Gambar 3. Tahap integrasi

2. Cleansing

Cleansing adalah tahapan pembentukan model standar dari *schema* data nasabah. Model dibuat dengan mendefinisikan setiap kolom yang akan digunakan dari data hasil integrasi yang didapatkan pada proses sebelumnya. Skema modeling *single* CIF dapat dilihat pada Tabel 1.

Tabel 1. Skema model nasabah individu

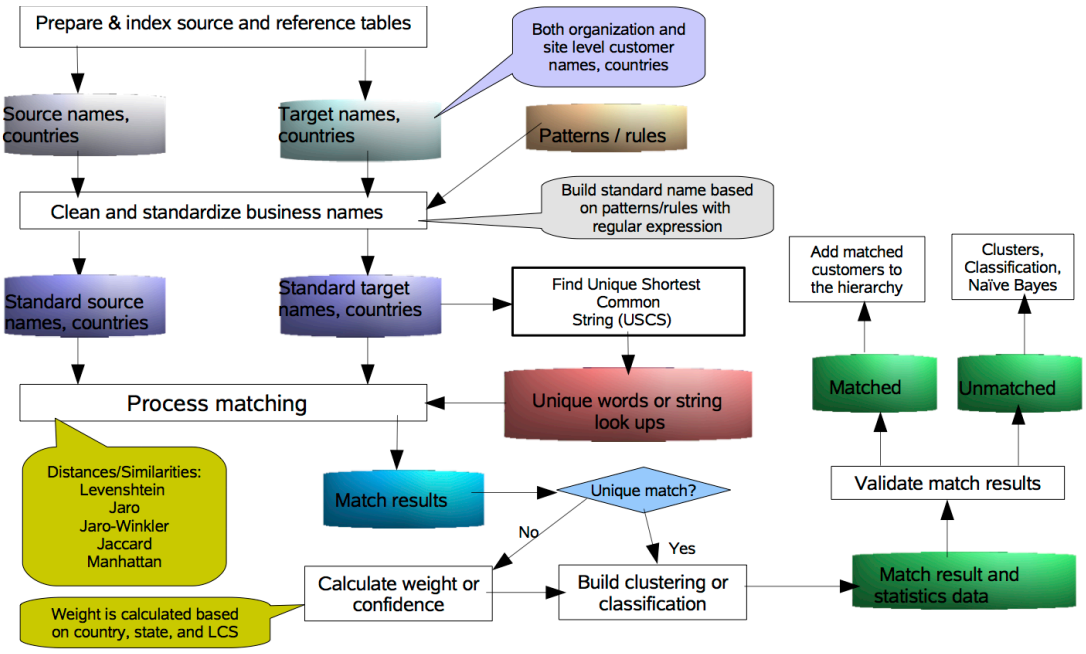
No	Personal	Tipe
1	ID Number	Fuzzy
2	Nama	Fuzzy
3	Tanggal Lahir	Exact
4	Alamat	Fuzzy
5	Ibu Kandung	Fuzzy
6	NPWP	Exact
7	Tempat Lahir	Fuzzy
8	Email	Fuzzy
9	Phone number	Fuzzy
10	Maintenance Date	Exact

Setelah model ditetapkan, dilakukan proses *cleansing content by parameter*. Proses *cleansing* pada intinya adalah pembersihan tanda baca, membuat standardisasi data dan melakukan proses *enrichment* dengan metode *find-what* dan *replace*. Proses lengkapnya sebagai berikut:

a. *Removing punctuation*
   Menghapus tanda baca seperti: titik, koma, titik-koma, *slash*, *back-slash*, *plus*, *minus*, tanda kurung, tanda seru, tanda tanya, tanda kurung siku dsb.
b. *Standardization*
   - Standarisasi Penulisan:
     i. Jika ada gelar, maka akan dihapus atau dipindah ke sebuah kolom baru
     ii. Penulisan Jalan atau Jl. atau Jln.
     iii. Pemisahan Nama Jalan, RT, RW, Komplek, Desa, Kecamatan, Kabupaten/Kota
     iv. Pemisahan kode wilayah ke kolom tersendiri
     v. Memperbaiki alamat email yang memiliki salah format
c. *Enrichment*
   Setelah standarisasi, data dapat dilengkapi melalui proses *find-what* dan *replace-by*.

3. Matching

Tahapan berikutnya adalah *matching proccess* atau pencocokan data. Skema pada Gambar 4 menggambarkan alur pencocokan data. Skema tersebut dimulai dengan menentukan sumber data dan target data. Kemudian dilakukan pembersihan data (pada tahap *cleansing*). Setelah dihasilkan data standar, pencocokan dilakukan menggunakan *fuzzy matching* dengan algoritme perhitungan jarak *string* seperti algoritme **Levenshtein**. Data yang cocok satu sama lain diberikan nilai dan pembobotan. Pada tahap akhir, skema tersebut memberikan luaran berupa data nasabah disertai dengan dua nilai yaitu **_score** dan **_weighted_score**. Dengan mengatur nilai *threshold* dari dua nilai tersebut, akan dihasilkan kelompok-kelompok data.



Gambar 4. Skema *mathcing*

Secara matematis, perhitungan jarak **Levenshtein** antara dua *string* a,b (dari masing-masing panjang | a | dan | b |) diberikan oleh formula $\mathbf{Lev_{a,b}}$ (| a |, | b |),

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Misalnya, jarak Levenshtein antara "kitten" dan "sitting" adalah 3, karena ada tiga cara pergantian karakter untuk mengubah *string* ke-1 menjadi *string* ke-2 sebagai berikut:

I. Kitten → sitten (substitusi dari "s" untuk "k")
II. Sitten → sittin (pengganti "i" untuk "e")
III. Sittin → duduk (penyisipan "g" di bagian akhir).

Selain itu, akan diterapkan juga jarak **Damerau–Levenshtein** sebagai perbandingan metode. Secara matematis, perhitungan jarakn antara dua *string* a,b sesuai fungsi $\boldsymbol{d_{a,b}(i,j)}$ di bawah ini:

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} d_{a,b}(i-1,j) + 1 \\ d_{a,b}(i,j-1) + 1 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \\ d_{a,b}(i-2,j-2) + 1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ \min \begin{cases} d_{a,b}(i-1,j) + 1 \\ d_{a,b}(i,j-1) + 1 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

4. Grouping

Grouping adalah tahapan pengelompokan data berdasarkan nilai **_score** dan **_weighted_score**. **_score** adalah nilai kecocokan atau *similarity* sebuah *record* dengan *record pivot*. *Record pivot* adalah *record* yang dipilih untuk dibandingkan dengan *record-record* lainnya. Nilai **_score** dihitung dengan membandingkan kecocokan *string* dari setiap kolom atau kolom yang menjadi *matching-parameter* yang bertipe *string*. Sementara **_weighted_score** adalah nilai *similarity* dari setiap kolom yang menjadi *matching-parameter* setelah dikalikan dengan bobotnya. Dengan mengatur *threshold* dari dua nilai tersebut, akan dihasilkan tiga kelompok data:

1. Data-Mathced: yakni kelompok data yang berisi data yang sudah dapat dipastikan merupakan satu group dan dapat ditentukan *golden* datanya
2. Data-Steward: yakni kelompok data yang berisi data yang ditemukan mirip namun tidak memenuhi syarat sebagai *matched group*
3. Data-Unmatched: yakni kelompok data yang berisi data yang sudah dapat dipastikan tidak memiliki *group* atau bisa juga disebut *single* data

Nilai *treshold* terdiri atas nilai ***upper threshold*** dan nilai ***lower threshold***, baik **_score** maupun **_weighted_score.** Sebagai contoh, kelompok Data-Matched akan memiliki persyaratan:

(_score >= *upper_score* && _weighted_score >= *upper_weight_score*)

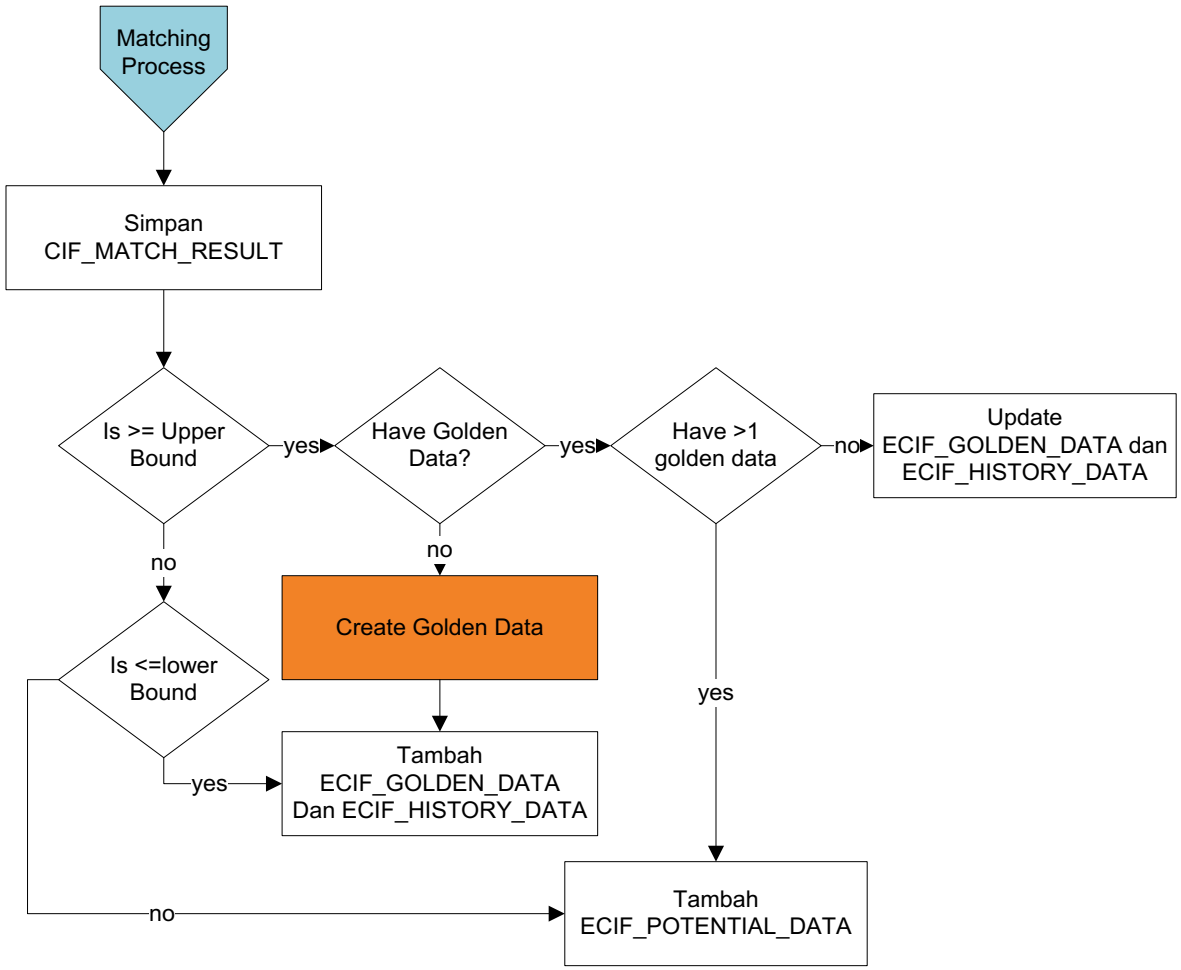
Kelompok Data-Stewardess memiliki syarat:

(_score >= *lower_score* && _score < *upper_score*) && (_weighted_score >= *lower_weight_score* && _weighted_score < *upper_weight_score*)

Kelompok Data-Unmatched memiliki persyaratan:

(_score < *lower_score* && _weighted_score < *lower_weight_score*)

Setelah parameter ***treshold*** ditetapkan, dilakukan proses pembangkitan *golden data* yang disebut ***Generates Golden Data Automatically*** sesuai alur pada Gambar 5.



Gambar 5. *Generates golden data automatically*

Setiap *record* nasabah yang nilai ***score*** *dan* ***weight_*****score** diatas atau sama dengan nilai ***upper treshold***, maka data tersebut akan dicek apakah memiliki *golden data* atau tidak. Jika tidak, maka akan masuk proses *create golden data*. Jika memiliki *golden data*, maka *record* tersebut akan masuk ke dalam *history golden data* dan memperbaharui *golden data*. Selain itu, untuk *record* nasabah yang telah memiliki *golden data* lebih dari satu dan memiliki nilai ***score*** dan ***weight_*****score** terletak diantara ***upper bound treshold*** dan ***low bound treshold*** akan menjadi *potential data*. Potensial data akan diproses menjadi *golen data* pada aplikasi E-CIF berbasis *website* melalui penilaian manusia. Sedangkan *record* yang memiliki nilai ***score*** dan ***weight_*****score** dibawah ***low bound treshold*** akan menjadi *golden data* baru dan *record* tersebut akan menjadi *history* bagi *golden data*-nya sendiri.

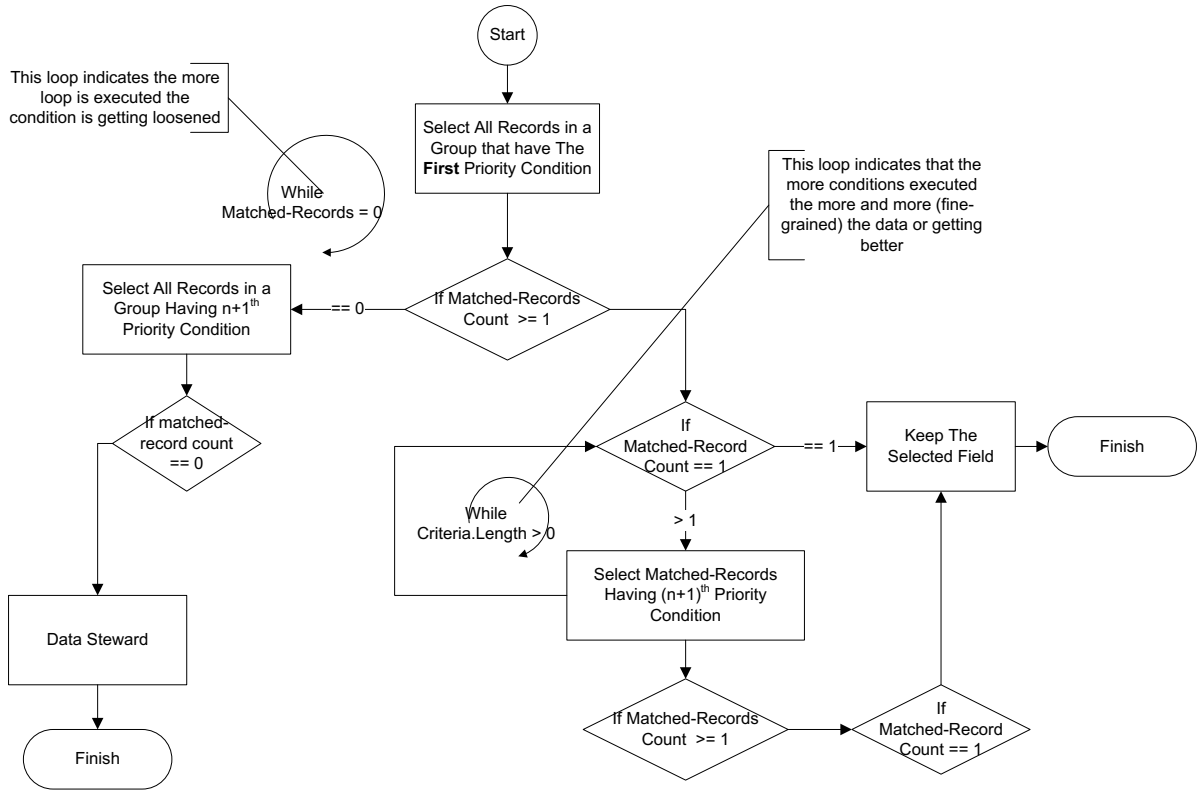
Algoritma *Create Golden Data*

Sebelum menjalankan proses *algoritma create golden data*, dibuat dahulu aturan sebagai acuan pembentukan *golden data.* Aturan dapat dimodifikasi sesuai kebutuhan pembentukan *golden data.* Ilustrasi aturan dapat dilihat pada Gambar 6.

	F1	F2	F2p	F3	F3p	F4	F4p	F5	F5p	F6	F6p	F7	F7p	F8	F8p	F9	F9p
No.	FieldName	Taken From Source	Priority	Is From Record Having Most Updated (last Maintenace Date) Row	Priority	Follow Regex	Priority	The Field taken from Most Recent Date (for Date DataType Only)	Priority	Field which has The Longest String Value	Priority	Has more number of the same value	Priority	Similar to Domain	Priority	Take Not NULL Regardless of Source	Priority
1.	CustNm	NCBS	1	TRUE	2	NULL	NULL	FALSE	NULL	TRUE	4	TRUE	3	NULL	NULL	TRUE	5
2.	IdNoKTP	NCBS	1	TRUE	2	[0..9]*	3	FALSE	NULL	FALSE	NULL	TRUE	4	NULL	NULL	TRUE	5
3.	BirthDate	NCBS	1	TRUE	2	NULL	NULL	FALSE	NULL	FALSE	NULL	TRUE	3	NULL	NULL	TRUE	4
4.	BirthPlace	NCBS	1	TRUE	2	NULL	NULL	FALSE	NULL	FALSE	NULL	TRUE	3	CITY	4	TRUE	5
5.	Alamat	NCBS	1	TRUE	2	NULL	NULL	FALSE	NULL	TRUE	4	TRUE	3	NULL	NULL	TRUE	5
6.	MotherName	NCBS	1	TRUE	2	NULL	NULL	FALSE	NULL	TRUE	4	TRUE	3	NULL	NULL	TRUE	5
7.	MaintenaceDt	NULL	NULL	FALSE	NULL	NULL	NULL	TRUE	1	FALSE	NULL	FALSE	NULL	NULL	NULL	TRUE	2

Gambar 6. Aturan pembentukan *golden data*

Berikut alur algoritma *create golden data* dapat dilihat pada Gambar 7. Algrotima tersebut berjalan dengan mengikuti aturan yang didefinisikan pada *user interface* pada Gambar 6.



Gambar 7. Algoritma *create golden data*

Untuk setiap *highly-confidence group-data* atau data dengan **_key_out** yang sama dan memiliki **_score** tinggi dan **_weighted_score** tinggi dilakukan proses seleksi mengikuti aturan pada Gambar 6 untuk membuat *golden data*. Proses seleksinya sebagai berikut:

1. Seleksi semua *records* yang memenuhi kriteria dengan prioritas = 1 (*select all records in a group that have the first priority condition*). Kriteria prioritas tertinggi (1) adalah F3, yaitu harus diambil dari *record* terbaru.

2. Jika hasil proses seleksi prioritas pertama di atas menghasilkan *records* dengan nilai *kolom* LastUpdated yang sama (*if matched-records count >= 1*) maka perlu dilanjutkan dengan pengetatan kriteria dengan kriteria ke-2. Jika hasil seleksi prioritas pertama

menghasilkan *record* sejumlah >= 1, maka perlu dilakukan penerapan kriteria ke-2.

3. Jika hasil proses seleksi prioritas pertama sudah menghasilkan sebuah record (*if matched-record count == 1*), maka *kolom* (dalam contoh ini, CustNm) akan diambil dari *record* tersebut *(keep selected column*) dan proses selesai.

4. Proses seleksi selanjutnya adalah melakukan pengetatan kriteria, jika ternyata hasil seleksi prioritas pertama masih menghasilkan banyak *records*, maka dilakukan dengan prioritas ke-2 dan seterusnya sedemikian sehingga didapat 1 *record* (bisa juga beberapa *record,* namun nilai kolomnya sama) yang akan dijadikan *golden data* atau sampai jumlah kriteria habis terseleksi (*while criteria.Length > 0*).

5. Pada contoh kasus CustNm di atas, jika kriteria pertama menghasilkan 3 *records* dengan *last-update* yang sama, maka akan dipilih dari 3 *records* itu yang memiliki frekuensi *occurrences*-nya lebih banyak.

6. Jika ada dua dengan nama yang sama, maka CustNm dengan dua *occurrences* inilah yang akan menjadi *golden data*. Jika semua CustNm memiliki isian yang mirip, maka harus dilanjutkan dengan kriteria ke-3.

7. Kriteria ke-3, yaitu dari ketiga itu harus dipilih yang paling panjang stringnya. Jika tidak ditemukan juga yang paling panjang, maka gunakan kriteria ke-4.

8. Kriteria ke-4, yakni apakah ada *record* yang berasal dari NCBS, jika ada *golden data* didapat, jika tidak maka hanya karena satu kolom ini saja (CustNm) *golden data* tidak bisa dibentuk.

9. Pembentuk *golden data* akan melalui *merge* atau *unmerge data-steward* pada tahap evaluasi *end-user* ECIF.

5. Evaluation

Tahapan terakhir adalah *evaluation* atau tahapan verifikasi dan koreksi *golden data* atau *single* CIF. Evaluasi dilakukan pada aplikasi website E-CIF. Evaluasi akan dilakukan oleh pihak BDI sebagai pakar untuk menghasilkan data *single* CIF terbaik. Skenario ini secara umum melakukan *merge*, *unmerge,* dan *update* pada *golden data* dan *potensial data* untuk menentukan *parent* CIF dan *child* atau histori CIF.

Parent CIF merupakan *golden data* terbaik atau *single* CIF terbaik. Sedangkan, *child* merupakan histori data dari *golden data*. Histori data berisi CIF yang memiliki skor kemiripan yang tinggi dengan *golden data.*

Merge dilakukan jika ada data CIF yang masih dianggap mirip pada masing-masing *golden data* atau pun *potensial* data. *Merge* diinisiasi dengan memilih CIF yang dianggap mirip. Kemudian diterapkan sebuah aturan untuk menentukan pilihan isian terbaik dari *kolom*-kolom data pada CIF. Aturan tersebut mengikuti algoritme *fitering* CIF, yaitu isian terbaik ditentukan dengan memperhatikan: 1) updated_data terbaru, 2) ide_source prioritas, yaitu

NCBS, Ascend, Adira, 3) isian yang tidak *null* atau kosong. Setelah isian ditentukan, maka akan dilakukan *approval* untuk menyetujui CIF tersebut menjadi *single* CIF baru. Sedangkan, *unmerge* dilakukan jika ada data histori dari *single* CIF yang dianggap memiliki kemiripan yang rendah dengan *golden data* atau *parent.* CIF yang diproses *unmerge* akan membentuk *golden data* baru dengan histori CIF itu sendiri.

Pembaharuan atau *update golden data* dilakukan jika ada data eksternal yang dianggap oleh pakar adalah *golden data* atau data histori dari single CIF yang telah ada. Setiap proses *merge*, *unmerge* dan *update* pada *golden data* akan melalu proses approval.

Pada website E-CIF juga disediakan *user interface* untuk pengguna dalam menentukan parameter *matching,* menentukan parameter dan mekanisme *cleansing find-what* dan *replace-by*, menentukan parameter *treshold* untuk *upper bound* dan *lower bound* untuk *score dan weight_score* dan menentukan mekanisme level approval.

Jadwal Penelitian

Waktu yang diperlukan dalam menyelesaikan penelitian ini selama 6 bulan dan untuk lebih lengkapnya seperti yang terlihat pada Tabel 2.

Tabel 2 Jadwal Penelitian

No	Rencana Penelitian	Tahun/Bulan							
		2017		2018					
		Ags	Sep	Okt	Nov	Des	Jan	Feb	Mei
1	Studi Literatur	■	■						
2	Sidang Komisi I		■						
3	Pendaftaran Kolokium dan Proposal			■	■				
4	Kolokium				■				
5	Praproses Data		■	■	■	■			
6	Penyusunan Draft Tesis a. Pendahuluan b. Tinjauan Pustaka c. Metodologi		■	■	■				
7	Analisis dan Penerapan Metode					■			
8	Evaluasi dan Validasi Hasil					■	■		
9	Publikasi Ilmiah					■			
10	Penyusunan Draft Tesis d. Hasil dan Pembahasan e. Kesimpulan f. Saran						■	■	
11	Seminar							■	
12	Ujian Akhir								■

Daftar Pustaka

Andrejková Gabriela *et al.* 2013. *Approximate Pattern Matching using Fuzzy Logic*. ITAT 2013 Proceedings, CEUR Workshop Proceedings. 1003:52–57.

Beall Scott K and Hodges Robert L. 2003. *Customer Information File Management: Software Comparison Columns.* DPRO-90048.

Gupta Gaurav and Aggarwal. 2012. *Improving Customer Relationship Management Using Data Mining.* International Journal of Machine Learning and Computing. 2:874-877.

Jian Yu and Miin-Shen Yang. 2007. *A Generalized Fuzzy Clustering Regularization Model With Optimality Tests and Model Complexity Analysis*. IEEE Transactions on Fuzzy Systems. 5: 904-915.

Kolb Lars *et al.* 2009. *Dedoop: Efficient Deduplication with Hadoop*. Proceedings of the VLDB Endowment. 5:1-12.

Lam Vincent and Taylor JT. 2009. *Enterprise Information Management (EIM): The Hidden Secret to Peak Business Performance*. Information Builders. 8:1-8.

Prabha S. 2014. *Reduction Of Big Data Sets Using Fuzzy Clustering*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 3:2235-2238.

Reverleen House. 2011. *Exploiting the Single Customer View to Maximise the Value of Customer Relationships*. Experian. 1:1-14.

Thompson *et al*. 2003. *Enterprise information management system and methods*. United State Patent: US 6668253 B1.

Verhoef, Peter C and Bas Donkers. 2001. *Predicting Customer Potential Value An Application In The Insurance Industry.* Erasmus Research Institute of Management (ERIM). 200:1-35.