

**PROPOSAL TUGAS AKHIR
UNIVERSITAS BAKRIE
TAHUN 2015**

**RANCANG BANGUN SISTEM *DATA CLEANING* UNTUK
MASTER DATA KONSUMEN DI PT XYZ DENGAN
MENERAPKAN METODE *SORTED NEIGHBOURHOOD* DAN
METODE *N-GRAM***

**Fakultas Teknik dan Ilmu Komputer
Program Studi Informatika**

**Rahma Mualifa
NIM: 1112001011**



**Universitas Bakrie
Kampus Kuningan Kawasan Epicentrum
Jl HR Rasuna Said Kav. C-22, Jakarta 12920**

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : Rahma Mualifa
NIM : 1112001011
Program Studi : Informatika
Fakultas : Teknik dan Ilmu Komputer
Judul Skripsi : Rancang Bangun Sistem *Data Cleaning* Untuk Master Data Konsumen di PT XYZ Dengan Menerapkan Metode *Sorted Neighbourhood* dan *N-Gram*

Telah diseminarkan dan disetujui oleh pembimbing dan pembahas tugas akhir untuk diajukan ke sidang tugas akhir.

Jakarta, 30 Juni 2016

Menyetujui,

Pembimbing Tugas Akhir,

Pembahas Tugas Akhir,

Yusuf Lestanto, S.T., M.Sc.

Guson P. Kuntarto S.T., M.Sc.

Rancang Bangun Sistem *Data Cleaning* Untuk Master Data Konsumen di PT XYZ Dengan Menerapkan Metode *Sorted Neighbourhood* dan *N-Gram*

Rahma Mualifa

ABSTRAK

Penelitian ini membahas tentang rancang bangun sistem *data cleaning* untuk dapat mendeteksi duplikasi data yang ada pada master data konsumen Divisi *Consumer Care* PT XYZ. Metode yang digunakan dalam penelitian ini untuk mendeteksi duplikasi data adalah dengan menerapkan pendekatan metode *Sorted Neighbourhood* (SNM) dan *N-Gram*. Sistem *data cleaning* ini bertujuan membantu *user* untuk dapat mempermudah menemukan duplikasi data. Selain itu, sistem ini juga dapat membantu *user* untuk dapat merapikan format penulisan telepon dan fax yang ada pada master data konsumen Divisi *Consumer Care* PT XYZ. Sistem yang akan dibangun adalah sistem *web based* dengan menggunakan bahasa pemrograman C#. Hasil dari sistem *data cleaning* yang dibangun kemudian akan diuji coba kepada *user* dan dinilai seberapa efektif metode SNM dan N-Gram dalam mendeteksi duplikasi data dengan menghitung nilai *recall* dan *precision* terhadap hasil proses deteksi duplikasi data.

Kata kunci: *Data cleaning*, Deteksi Duplikasi Data, *Sorted Neighbourhood*, *N-gram*

Daftar Isi

ABSTRAK	iii
Daftar Isi.....	iv
Daftar Gambar.....	vi
Daftar Tabel	vii
Daftar Singkatan.....	viii
1 Pendahuluan	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan	4
2 Tinjauan Pustaka	6
2.1 Penelitian Terkait	6
2.2 <i>Data Cleaning</i>	8
2.3 Metode <i>Data Cleaning</i>	9
2.3.1 Algoritma Deteksi Duplikasi Data	10
2.3.2 Metode <i>Sorted Neighbourhood</i> Sebagai Metode Untuk Deteksi Duplikasi Data	11
2.3.3 Algoritma Perhitungan Kemiripan Antar String	12
2.3.4 Algoritma Pendekatan <i>N-Gram</i> Sebagai Algoritma Perhitungan Kemiripan Antar <i>String</i>	13
2.4 Pemrograman Berorientasi Objek	14
2.5 <i>Unified Modelling Language</i> (UML).....	15
3 Metodologi Penelitian	19
3.1 Tahap Identifikasi Masalah	19
3.1.1 Prosedur Yang Sedang Berjalan	20
3.1.2 Master Data Konsumen Divisi <i>Consumer Care</i> di PT XYZ.....	21
3.1.3 Struktur Organisasi	23

3.1.4	Bisnis Proses.....	25
3.1.5	Sistem <i>Data Cleaning</i> Yang Diajukan	26
3.2	Tahap Analisa Kebutuhan Sistem	27
3.2.1	Kebutuhan Non Fungsional Sistem	27
3.2.2	Analisa Kebutuhan Fungsional Sistem.....	27
3.3	Perancangan Sistem	27
3.4	Tahap Implementasi	28
3.5	Tahap Pengujian.....	28
3.6	Jadwal Penelitian.....	29
	Daftar Pustaka	30
	Lampiran 1 – Wawancara	33
	Lampiran 2 – <i>Requirement Elicitation</i>	37

Daftar Gambar

Gambar 3.1 Struktur Organisasi Divisi <i>Consumer Care</i> PT XYZ.....	23
Gambar 3.2 Gambar Bisnis Proses Deteksi Data Kembar Pada Master Data Konsumen Divisi <i>Consumer Care</i> PT XYZ.....	26

Daftar Tabel

Tabel 2.1 Perbandingan Metode dari Beberapa Penelitian Terkait	7
Tabel 2.2 Tabel Simbol-Simbol Pada <i>Use Case Diagram</i>	16
Tabel 2.3 Tabel Simbol-Simbol Pada <i>Activity Diagram</i>	17
Tabel 2.4 Tabel Simbol-Simbol Pada <i>Class Diagram</i>	18
Tabel 3.1 Kamus Data Konsumen Divisi <i>Consumer Care</i> PT XYZ	21
Tabel 3.2 Tabel <i>Role</i> dan Deskripsi Kerja Divisi <i>Consumer Care</i> PT XYZ	23
Tabel 3.3 Jadwal Penelitian.....	29

Daftar Singkatan

IDE	Integrated Development Environment
IGASIS	Intra-Governmental Access To Shared Information System
KDD	Knowledge Discovery in Databases
OOP	Object Oriented Programming
SNM	Sorted Neighbourhood Method
UML	Unified Modelling Language

Bab I

Pendahuluan

1.1 Latar Belakang Masalah

Data merupakan hal yang sangat penting untuk dapat menghasilkan sebuah informasi. Data yang memiliki jumlah besar biasanya dimiliki oleh beberapa organisasi/perusahaan dengan proses bisnis yang sangat kompleks. Data yang besar ini berasal dari serangkaian proses bisnis untuk digunakan sebagai proses pengambilan keputusan. Tepat atau tidaknya sebuah organisasi/perusahaan untuk mengambil sebuah keputusan bergantung pada kualitas data yang organisasi/perusahaan miliki. Namun, data yang berasal dari sumber eksternal organisasi/perusahaan biasanya memiliki data kotor. Hal ini biasanya terjadi karena kualitas data yang rendah seperti adanya duplikasi data, penulisan ejaan yang salah, atau data yang tidak lengkap. Data kotor inilah yang menyebabkan hasil laporan sebuah organisasi/perusahaan menjadi tidak akurat sehingga akan menyebabkan suatu kesalahan keputusan dalam sebuah organisasi/perusahaan (Guo, dkk., 2012).

Data kotor yang muncul dapat terjadi karena beberapa alasan (meskipun sebuah organisasi/perusahaan hanya memiliki satu atau *single database*). Kesalahan ejaan pada saat proses memasukkan data dan penulisan yang tidak memiliki standar format merupakan penyebab munculnya data kotor. Selain itu, data baru yang ternyata sudah ada di dalam *database* dan kemudian dimasukkan kembali ke dalam *database* menyebabkan *database* menjadi memiliki duplikasi data. Terlebih lagi, jika suatu organisasi/perusahaan memiliki beberapa sumber data yang heterogen sehingga adanya perbedaan model data antara sumber data yang satu dengan yang lain (Couto, 2012).

PT XYZ merupakan salah satu perusahaan farmasi yang memiliki puluhan ribu data dalam *database* konsumen yang dimilikinya. Data konsumen yang dibahas dalam penelitian ini adalah data konsumen yang berada pada divisi *Consumer Care* yang ada pada PT XYZ. Data konsumen yang ada pada divisi

tersebut masih memiliki kualitas data yang rendah karena terdiri atas data yang berduplikasi dan terdapat beberapa data yang belum mempunyai standar format penulisan, khususnya penulisan nomor telepon dan fax.

Berdasarkan wawancara yang telah penulis lakukan terhadap staf *sales admin* yang mengelola data konsumen di Divisi *Consumer Care* PT XYZ, telah diketahui bahwa proses pembersihan data, yaitu berupa deteksi duplikasi data dan merapikan format telepon dan fax masih dilakukan secara manual dan membutuhkan waktu yang lama. Hal ini karena jumlah data konsumen berjumlah ± 25.000 *record*. Sedangkan, jumlah data yang berduplikasi di dalam data tersebut berkisar ± 1.300 *raw* data atau terdiri atas 5.2% data yang berduplikasi. Oleh karena itu, dibutuhkan tingkat akurasi yang stabil untuk mendeteksi duplikasi data yang ada di dalam master data konsumen PT XYZ.

Berdasarkan permasalahan yang ada di Divisi *Consumer Care* PT XYZ, penulis bermaksud menerapkan suatu sistem *data cleaning* untuk dapat mendeteksi duplikasi data yang ada pada data konsumen PT XYZ secara otomatis. Namun, untuk dapat menerapkan sistem tersebut dibutuhkan suatu metode deteksi duplikasi data untuk dapat membantu permasalahan Divisi *Consumer Care*. Metode pendekatan yang digunakan untuk membangun sistem *data cleaning* ini adalah dengan menerapkan Algoritma *Sorted Neighbourhood* dan menggunakan Algoritma *N-gram*.

Algoritma *Sorted Neighbourhood* merupakan algoritma untuk mendeteksi duplikasi data dengan membentuk token khusus lalu kemudian menggabungkan dan menghapus dua buah atau lebih data yang kembar (Hernandez dan Stolfo, 1995). Sedangkan, Metode *N-gram* merupakan salah satu metode untuk menghitung kemiripan antar *string* yang menjadi penentu apakah antar *record* merupakan *record* yang kembar atau tidak (Recchia dan Max, 2013). Kedua metode ini akan diterapkan dalam penelitian ini untuk dapat menemukan duplikasi data pada master data konsumen Divisi *Consumer Care* PT XYZ.

Berdasarkan uraian di atas, maka dalam penelitian ini penulis bermaksud merancang suatu sistem *data cleaning* untuk master data konsumen pada PT XYZ dengan memanfaatkan Algoritma SNM dan Metode *N-Gram* dengan judul

penelitian “Rancang Bangun Sistem *Data Cleaning* Untuk Master Data Konsumen di PT XYZ Dengan Menerapkan Metode *Sorted Neighbourhood* dan Metode *N-Gram*”.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah, disusun perumusan masalah sebagai berikut.

1. Bagaimana mengembangkan suatu sistem *data cleaning* untuk master data konsumen Divisi *Consumer Care* PT XYZ agar dapat menerapkan algoritma SNM dan N-Gram.
2. Langkah yang digunakan untuk mengukur nilai efektivitas dari hasil deteksi duplikasi data pada sistem *data cleaning* yang dibangun.

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah, disusun tujuan penelitian sebagai berikut:

1. Mengembangkan sistem *data cleaning* untuk dapat menerapkan Algoritma *Sorted Neighbourhood* dan Algoritma *N-Gram* agar dapat mendeteksi duplikasi data yang ada pada master data konsumen Divisi *Consumer Care* PT XYZ.
2. Mengukur nilai efektivitas hasil deteksi duplikasi data terhadap sistem *data cleaning* yang dibangun untuk master data konsumen Divisi *Consumer Care* PT XYZ.

1.4 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

- a. Bagi PT XYZ:
 1. Sistem *data cleaning* yang dibuat akan digunakan sebagai *tools* untuk menyeleksi duplikasi data dan merapikan format penulisan telepon dan fax yang ada pada data konsumen Divisi *Consumer Care* PT XYZ.

2. Waktu yang digunakan oleh PT XYZ dalam proses mendeteksi duplikasi data menjadi lebih cepat dibanding dengan metode konvensional yang biasa dilakukan oleh staf *Sales Admin* Divisi *Consumer Care* PT XYZ.
- b. Bagi Universitas Bakrie:
 1. Hasil penelitian dapat dijadikan sebagai dokumen akademik yang dapat dijadikan sebagai bahan literatur bagi sivitas akademika Universitas Bakrie.

1.5 Batasan Masalah

Batasan ruang lingkup dari penelitian yang dibahas dalam penelitian ini adalah sebagai berikut:

- a. Masalah yang diteliti adalah tentang sistem *data cleaning* yang terbatas untuk master data konsumen yang ada pada Divisi *Consumer Care* PT XYZ.
- b. Implementasi algoritma *Sorted Neighbourhood Method* dan *N-Gram* diterapkan dalam sistem untuk mendeteksi duplikasi data pada data konsumen Divisi *Consumer Care* PT XYZ.
- c. Sistem *data cleaning* yang akan dibangun hanya terbatas untuk mendeteksi duplikasi data dan merapikan format penulisan telepon dan *fax*.
- d. Sistem yang akan dibangun merupakan sistem berbasis *web*.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini adalah sebagai berikut:

- a. Bab I Pendahuluan
Pada bab ini dijelaskan mengenai latar belakang penelitian, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan.
- b. Bab II Landasan Teori

Pada bab ini dibahas mengenai dasar-dasar teori, rujukan dan metode yang digunakan sebagai dasar dan alat untuk menyelesaikan permasalahan.

c. Bab III Analisis dan Perancangan Sistem

Pada bab ini dijelaskan tentang analisis serta perancangan sistem *data cleaning* untuk master data konsumen Divisi *Consumer Care* PT XYZ.

d. Bab IV Implementasi Program dan Pengujian

Pada bab ini berisi penerapan program dan pengujian sistem *data cleaning* yang dibangun telah sesuai dengan kebutuhan *Sales Admin* Divisi *Consumer Care* PT XYZ atau tidak.

e. Bab V Simpulan dan Saran

Pada bab ini berisi tentang simpulan dari hasil pembuatan sistem *data cleaning* dan saran-saran yang ditujukan kepada semua pihak yang bersangkutan.

Bab II

Tinjauan Pustaka

2.1 Penelitian Terkait

Pada tahun 1995, Hernandez dan Stolfo melakukan penelitian terhadap metode terkait *data cleaning* dengan penelitiannya yang berjudul *The Merge/Purge Problem for Large Databases*. Dalam penelitian ini, dijelaskan tentang Metode *Sorted Neighbourhood Method* (SNM) dan Metode SNM dengan *Clustering* data terlebih dahulu sebagai metode untuk menyelesaikan masalah penggabungan atau penghapusan dua buah atau lebih data yang kembar. Kemudian, Hernandez dan Stolfo juga membandingkan dua buah metode tersebut dan ternyata Metode SNM yang menerapkan tahap *clustering* data terlebih dahulu lebih memiliki performa yang lebih baik dibandingkan dengan Metode SNM.

Pada tahun 1999, Lee, dkk. melakukan penelitian terhadap metode terkait *data cleaning* dengan penelitiannya yang berjudul *Cleansing Data for Mining and Warehousing*. Dalam penelitian ini, metode *Sorted Neighbourhood Method* (SNM) dengan sedikit pengembangan digunakan untuk mendeteksi adanya *record* yang kembar. Dalam penelitian ini, dilakukan beberapa langkah untuk melakukan proses *data cleaning*, yaitu (1) pembersihan *field* dari data kotor, (2) melakukan proses tokenisasi dan mengurutkan token yang ada pada *field*, (3) mengurutkan *record*, (4) membandingkan *record*, dan (5) menggabungkan dua atau lebih *record* yang sama menjadi satu *record*.

Selanjutnya, penelitian berikutnya adalah penelitian yang dilakukan oleh Tian, dkk. pada tahun 2001. Penelitian tersebut membahas tentang metode algoritma *n-gram* untuk proses *data cleaning*. Dengan menggunakan pendekatan *n-gram*, setiap *record* dihitung jumlah nilai *n-gram* berdasarkan total dari nilai *n-gram* dari setiap *string* yang ada pada *record*. Jumlah angka tiap *record* kemudian dikelompokkan. *Record* yang berada dalam kelompok yang sama adalah kelompok yang terdeteksi memiliki data yang duplikat.

Kemudian, penelitian pada tahun 2006 dilakukan oleh Azma mengenai penerapan *data cleaning* dalam *data warehouse* sistem IGASIS (*Intra-Governmental Access To Shared Information System*). Proses pembersihan data yang dilakukan menerapkan metode pendekatan *schema matching* untuk membantu *user* atau *domain expert* dalam proses membersihkan dan memetakan data untuk dimasukkan ke *data warehouse*. Kemudian, menerapkan metode *linguistic-matching* dengan menggunakan metode *n-gram* untuk mengukur nilai kesamaan dari dua buah *string*.

Secara ringkas, penulis merangkum penelitian terkait pada tabel 2.1 di bawah ini.

Tabel 2.1 Perbandingan Metode dari Beberapa Penelitian Terkait

Judul	Pengarang	Tahun	Metode	Penelitian Yang Dilakukan
<i>The Merge/Purge Problem for Large Databases</i>	Hernandez, Mauricio A dan Stolfo, Savatore J.	1995	Metode SNM standar dan Metode SNM dengan teknik <i>Clustering</i>	Menerapkan metode SNM sebagai metode untuk melakukan penggabungan atau penghapusan dua buah data kembar. Kemudian, mengajukan teknik <i>Clustering</i> sebagai metode yang lebih baik dibandingkan metode SNM.
<i>Cleansing Data for Mining and Warehousing</i>	Lee, Mong Li; Lu, Hongjun; Ling, Tok Wang; Ko, Yee Teng	1999	Metode <i>Sorted Neighbourhood Method</i> (SNM)	Menerapkan metode SNM untuk mendeteksi duplikasi <i>record</i> dan mengajukan beberapa metode pra-pemrosesan agar <i>record</i> yang sama berada di posisi yang berdekatan.
<i>An n-gram-based approach for detecting approximately duplicate database records</i>	Tian, Zengping; Lu, Hongjun; Ji, Wenyun; Zhou, Aoying; Tian, Zhong	2001	Metode pendekatan <i>n-gram</i>	Mengajukan suatu metode pendekatan berbasis <i>n-gram</i> untuk mendeteksi duplikasi pada <i>record</i> .

Judul	Pengarang	Tahun	Metode	Penelitian Yang Dilakukan
Pembuatan Alat Bantu Dalam Proses <i>Data Cleaning</i> Pada <i>Intra-Governmental Access to Shared Information System</i> (IGASIS)	Azma, Syarifatul	2006	Metode pendekatan <i>schema matching</i>	Membangun suatu <i>tools</i> untuk memproses <i>data cleaning</i> untuk sistem IGASIS dengan menerapkan metode <i>schema-based</i> dalam memetakan data yang akan dimasukkan ke dalam <i>data warehouse</i> dan <i>instance-based</i> yang merupakan gabungan dari <i>linguistic matching</i> , <i>structural matching</i> , dan inputan <i>matches-mismatches</i> untuk mendeteksi duplikasi data.

2.2 Data Cleaning

Data Cleaning merupakan sebuah proses yang digunakan untuk menentukan data yang tidak akurat, tidak lengkap, atau data yang tidak jelas yang kemudian diperbaiki agar memiliki data yang berkualitas. Proses tersebut dapat terdiri atas pengecekan format, pengecekan kelengkapan, menghilangkan duplikasi atau kesalahan lain yang ada pada data (Chapman, 2005).

Menurut Maimon & Rokach (2006) dalam bukunya, *Data Mining and Knowledge Discovery Handbook*, *data cleaning* erat kaitannya dengan proses akuisisi dan definisi data untuk meningkatkan kualitas data yang ada pada sistem. *Data cleaning* merupakan bagian dari salah satu tahap awal proses *data mining*. *Data cleaning* juga biasa dikenal dengan sebutan *data scrubbing*, *data cleansing*, *error checking*, *error correction*, atau *error detection*.

Dalam proses *data mining*, *data cleaning* merupakan suatu tahap awal atau pra pemrosesan sejumlah data sebelum data diproses menjadi sebuah informasi atau pengetahuan. Sedangkan, dalam *data warehouse*, *data cleaning* didefinisikan sebagai bagian dari fase proses ETL (*extract*, *transform*, dan *load*) yang berfokus

pada proses deteksi dan koreksi terhadap data yang *error* yang bertujuan untuk meningkatkan kualitas data (Huda, 2010).

Salah satu tugas krusial dalam *data cleaning* atau *data scrubbing* adalah proses mendeteksi duplikasi data. Pada *database*, biasanya secara normal akan menghadapi permasalahan data seperti: (1) kesalahan atau kurang lengkapnya penulisan akibat *human error* ketika proses memasukkan data, (2) nilai yang dimasukkan tidak konsisten karena adanya perbedaan format ketika memasukkan data, (3) tidak lengkapnya informasi, (4) klien pindah dari satu tempat ke tempat lainnya tanpa ada pemberitahuan, dan (5) adanya kesalahan klien ketika memasukkan nama dan alamatnya (Lee, dkk., 1999).

Menurut Maletic dan Marcus (2000), bagaimanapun tidak ada definisi dan perspektif yang tepat yang diberikan terhadap proses *data cleaning*. Berbagai KDD (*Knowledge Discovery in Databases*) dan sistem *data mining* mengimplementasikan proses *data cleaning* dengan berbagai cara berdasarkan permasalahan dari kumpulan data yang ada.

2.3 Metode Data Cleaning

Secara umum, terdapat tiga langkah utama dalam proses *data cleaning*, yaitu:

1. Meng-audit data untuk mengidentifikasi jenis kesalahan yang mengurangi kualitas data,
2. Memilih metode yang cocok untuk mengotomatisasi pendeteksian dan penghilangan kesalahan, dan
3. Menerapkan metode tersebut pada *record* di dalam *dataset*.

Langkah (1) dan (2) dapat dilihat sebagai tahap spesifikasi dan tahap eksekusi dari alur kerja *data cleaning*. Sebagai tambahan, terdapat langkah selanjutnya, yaitu tahap *post-processing* atau kontrol dimana *user* dapat menguji hasil dan melakukan penanganan kesalahan terhadap hasil dari proses *data cleaning* (Maimon dan Rokach, 2005).

2.3.1 Algoritma Deteksi Duplikasi Data

Hernandez dan Stolfo (1995) dalam penelitiannya membahas tentang proses penggabungan atau penghapusan data dimana metode tersebut merupakan metode untuk menggabungkan duplikasi data dari dua atau lebih data yang kembar. Metode untuk mendeteksi duplikasi data tersebut diterapkan oleh penulis sebagai metode untuk membangun sistem *data cleaning* dalam penelitian ini. Berikut ini metode untuk mendeteksi duplikasi data menurut Hernandez dan Stolfo (1995).

1. Metode *Sorted Neighbourhood Method* (SNM)

Merupakan suatu metode pendekatan untuk membawa *record* yang berduplikasi berada pada posisi yang berdekatan. Lalu, proses deteksi duplikasi data dilakukan dalam ukuran *windows* tertentu untuk membatasi proses perbandingan satu data ke data lainnya. Memori yang dibutuhkan untuk memproses metode ini adalah $O(N \log N)$. Dimana N merupakan jumlah *record* di dalam database.

2. Metode SNM dengan teknik *Clustering*

Metode ini hampir serupa dengan metode SNM. Bedanya, data terlebih dahulu dibagi ke dalam beberapa *cluster* atau kelompok. Pembagian *cluster* atau kelompok dapat dilakukan secara independen sesuai dengan karakteristik data yang akan dibersihkan. Setelah data dikelompokkan, kemudian proses deteksi duplikasi data dilakukan pada tiap *cluster*. Memori yang dibutuhkan untuk memproses metode ini adalah $O(N \log \frac{N}{C})$. Dimana N merupakan jumlah *record* di dalam database dan C adalah ukuran *cluster*.

Berdasarkan pemaparan di atas, dapat disimpulkan bahwa Metode SNM dengan teknik *clustering* membutuhkan memori yang lebih sedikit dibandingkan dengan Metode SNM standar dimana $O(N \log \frac{N}{C}) < O(N \log N)$. Sehingga, pada penelitian ini akan diterapkan metode SNM dengan teknik *clustering* sebagai metode untuk mendeteksi duplikasi data.

2.3.2 Metode *Sorted Neighbourhood* Sebagai Metode Untuk Deteksi Duplikasi Data

Metode *Sorted Neighbourhood* yang akan digunakan pada penelitian ini adalah metode *Sorted Neighbourhood* dengan teknik *clustering* yang terdiri atas langkah berikut ini (Hernandez, 1995).

1. *Cluster data*. *Cluster data* dilakukan dengan menerapkan *constant partitioning* di mana data dikelompokkan ke dalam beberapa kelompok berdasarkan nilai yang ada pada atribut. Untuk mengelompokkan data dapat dilakukan dengan mengidentifikasi data yang ada. Dalam penelitian ini, data dapat dipisahkan berdasarkan atribut Area. Dengan demikian, proses perbandingan data dan proses pendeteksian data hanya dilakukan di tiap kelompok atau *cluster Area*.
2. Membentuk *key* atau token: Sebelum membentuk token, dilakukan tahap *pra-cleaning* terlebih dahulu seperti yang dilakukan oleh (Lee, dkk. 1999), yaitu berupa penghapusan kata, titel, tanda baca atau karakter tertentu. Untuk mendapatkan daftar karakter yang akan dihapus harus dilakukan observasi terlebih dahulu terhadap data yang akan dibersihkan. Setelah itu, baru dilakukan pembentukan token dengan mengambil satu atau beberapa huruf atau angka pada tiap kata yang ada pada tiap *field*. Misalnya, diambil dari tiga huruf pertama dari tiap *string* atau kata. Proses ini disebut dengan proses tokenisasi *record*.
3. Mengurutkan data: Mengurutkan *string* atau kata di dalam *field* dengan menggunakan *key* yang telah ditentukan pada tahap 1.
4. Menggabungkan data: sebuah *window* yang berukuran w bergerak melalui setiap *record* untuk membatasi proses perbandingan terhadap *record* yang berpotensi memiliki kesamaan data. Dimana nilai w adalah nilai jumlah pembagian tiap *cluster*. Setiap *record* baru yang masuk ke *window* dibandingkan dengan $w - 1$ *record* untuk menemukan *record* yang memiliki kesamaan.

2.3.3 Algoritma Perhitungan Kemiripan Antar String

Ketika antar *record* dibandingkan satu sama lain, dibutuhkan suatu metode untuk menghitung nilai kemiripan antar *string* yang ada di dalam *record* tersebut. Hal ini dilakukan untuk mengetahui apakah *record* tersebut sebenarnya termasuk *record* yang duplikat atau tidak. Berikut ini terdapat beberapa metode untuk menghitung kemiripan antar string menurut Recchia dan Max (2013):

1. *Edit Distance*

Algoritma ini mengukur banyaknya perbedaan antar *string* dalam hal jumlah adanya penyisipan, penghapusan, substitusi, dan/atau transposisi yang diperlukan untuk menghasilkan *string* pertama dari *string* kedua. Standar *Levenshtein Distance* menentukan nilai 1 untuk setiap penyisipan, penghapusan, dan substitusi. Kemudian, operasi tersebut dapat diubah menjadi nilai kemiripan antar *string* dengan membagi nilai aktual *Levenshtein Distance* dan nilai panjang *string* yang lebih panjang, dan mengurangi hasilnya dari nilai 1. Metode ini memiliki berbagai variasi yang mana variasi dari algoritma ini menentukan perbedaan bobot edit berdasarkan tipe operasinya dan pertimbangan lainnya (Navarro, 2001).

2. *Longest Common Substring* (LCS)

Metode LCS digunakan pada penelitian Friedman dan Sideli (1992) untuk mendeteksi dan membenarkan data pasien yang memiliki kesalahan ejaan dengan menentukan nilai LCS antar *string*. Nilai LCS didapat dengan menghitung pembagian antara jumlah *string* dengan posisi yang bersamaan dengan jumlah *string* terpendek atau jumlah *string* terpanjang ataupun rata-rata dari panjang kedua *string*.

3. *Smith-Waterman Distance*

Seperti *Edit Distance*, algoritma *Smith-Waterman* menentukan serangkaian operasi yang dibutuhkan untuk mentransformasikan dari satu *string* ke *string* lainnya. Nilai kemiripan *string* dapat diperoleh dengan

membagi antara hasil penghitungan *Edit Distance* dan panjang dari *string* terpanjang, *string* terpendek, atau rata-rata kedua *string*.

4. *N-Gram*

Metode ini mengukur banyaknya jumlah *n-gram* yang sama di antara dua *string* yang dibandingkan. Nilai kemiripan antar *string* didapat baik dengan membagi jumlah *n-gram* yang sama dengan jumlah *n-gram* di *string* yang pendek atau dengan jumlah *n-gram* di *string* yang panjang atau dengan rata-rata dari jumlah kedua *string* (Navarro, 2001).

Dari beberapa algoritma di atas, menurut Recchia dan Max (2013) dalam penelitiannya mengungkapkan bahwa algoritma *N-Gram* memiliki performa yang baik dari segi jumlah penemuan data (*recall*) dan ketepatannya (*precision*) dibandingkan dengan algoritma *Edit Distance*, *LCS*, dan *Smith-Waterman*. Oleh karena itu, penulis bermaksud untuk menerapkan metode *N-Gram* pada penelitian ini sebagai metode untuk menghitung nilai kemiripan antar string.

2.3.4 Algoritma Pendekatan *N-Gram* Sebagai Algoritma Perhitungan Kemiripan Antar *String*

Untuk membandingkan dua *record*, maka diterapkan algoritma pendekatan *n-gram* untuk mengukur nilai kesamaan dari dua *string* atau kata yang berbeda. Maksud dari *n-gram* adalah *n* huruf yang berturut-turut dari sebuah kata. Nilai *n* yang digunakan adalah 2, 3, dan 4. Jika *n* = 2, maka disebut digram atau bigram. Jika *n* = 3 disebut dengan trigram, dan seterusnya (Tian, dkk., 2001).

Berikut ini rumus *n-gram* untuk menghitung kemiripan antara *string* A dan *string* B:

$$Sim_{AB} = \frac{(2 (|ngram(A) \cap ngram(B)|))}{ngram(A) + ngram(B)}$$

Rumus 2.1 Rumus *N-Gram* Untuk Menghitung Kemiripan Antar *String*

Pengukuran nilai kemiripan antara dua *string* yaitu antara nilai 0 sampai 1. Jadi, semakin mirip *string* maka nilainya semakin mendekati 1. Sebaliknya, semakin tidak mirip nilainya akan mendekati 0. Berikut ini contoh penggunaan *n-gram* untuk menghitung nilai kemiripan antar *string*, dimana $n = 2$:

String A = “APOTIK”, mempunyai 2-gram:

AP, PO, OT, TI, IK

String B = APOTEK, mempunyai 2-gram:

AP, PO, OT, TE, EK

Dua contoh *string* di atas memiliki 3 buah bigram yang sama, yaitu AP, PO, dan OT sehingga nilai kemiripannya adalah:

$$Sim_{AB} = \frac{2 \times 3}{5+5} = 0,6$$

Untuk menentukan apakah dua *string* merupakan data yang kembar atau bukan maka akan dibuat ketentuan, yaitu dengan menentukan nilai ambang batas (*threshold*). Misalnya, untuk kasus ini, ditentukan nilai ambang batas = 0,6. Jadi, jika nilai kesamaan antara 2 *string* $\geq 0,6$, maka dapat dinyatakan bahwa 2 *string* tersebut memiliki kemiripan. Sebaliknya, jika nilai kesamaan antara 2 *string* $\leq 0,6$, maka dapat dinyatakan bahwa 2 *string* tersebut berbeda (Azma, 2006).

2.4 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah suatu pendekatan dalam pengembangan perangkat lunak di mana struktur perangkat lunaknya disusun berdasarkan objek-objek yang berinteraksi satu sama lain untuk menyelesaikan suatu tugas (Dan, 2011). Berikut ini adalah karakteristik dasar dari pemrograman berorientasi objek.

1. *Encapsulation* (Pemodulan/Pengkapsulan)

Metode untuk menggabungkan data dengan fungsi. Dalam konsep ini data dan fungsi digabung menjadi satu kesatuan yaitu kelas.

2. *Inheritance* (Pewarisan)

Dari konsep penurunan ini, suatu kelas bisa diturunkan menjadi kelas baru yang masih mewarisi sifat-sifat orang tuanya. Pewarisan dapat dilakukan jika:

- Ada beberapa atribut dan *method* yang sama yang digunakan oleh beberapa kelas berbeda (reduksi penulisan kode).
- Ada satu atau beberapa kelas yang sudah pernah dibuat yang dibutuhkan oleh aplikasi (*reusability*).
- Ada perubahan kebutuhan fungsional atau *feature* aplikasi dimana sebagian atau seluruh perubahan tersebut tercakup di satu atau beberapa kelas yang sudah ada (*extend*).

3. *Polymorphism* (Polimorfisme)

Polimorfisme berarti kelas-kelas yang berbeda tetapi berasal dari satu orang tua, dapat mempunyai metode yang sama tetapi cara pelaksanaannya berbeda. Atau dengan kata lain, suatu fungsi akan memiliki perilaku berbeda jika dilewatkan ke kelas yang berbeda-beda.

Jika dibandingkan dengan pemrograman secara prosedural, pemrograman berorientasi objek lebih memiliki keunggulan sebagai berikut (Ningsih, 2009).

1. Data dan fungsi dibungkus dalam kelas-kelas atau objek-objek sehingga dapat memudahkan pengembang aplikasi dalam memahami program.
2. Efektif digunakan untuk menyelesaikan masalah besar karena pemrograman berorientasi objek terdiri dari kelas-kelas yang memisahkan setiap *code* program menjadi kelompok-kelompok kecil sesuai dengan fungsinya.
3. Objek dan kelas dapat digunakan berkali-kali sehingga dapat menghemat *space* memori.

2.5 *Unified Modelling Language* (UML)


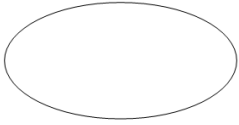

UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak

(Sulistyorini, 2009). UML menyediakan beberapa jenis diagram, diantaranya adalah sebagai berikut.

1. *Use Case Diagram*

Use case diagram adalah *diagram* yang menggambarkan interaksi antara sistem dan pengguna sistem. *Diagram* ini menjelaskan siapa yang akan menggunakan sistem dan memberikan sebuah narasi bagaimana *user* tersebut dapat berinteraksi dengan sistem. *Use case diagram* memiliki beberapa simbol antara lain:




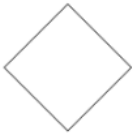


Tabel 2.2 Tabel Simbol-Simbol Pada *Use Case Diagram*

Nama Simbol	Notasi	Keterangan
<i>Actor</i>		Merepresentasikan pengguna sistem, tidak hanya manusia tetapi semua yang akan berinteraksi dengan sistem.
<i>Use Case</i>		Sebuah skenario (kegiatan) yang akan dilakukan untuk menyelesaikan suatu pekerjaan.
<i>Relationship</i>		Garis yang menghubungkan dua simbol pada <i>use case diagram</i> . Terdapat beberapa tipe <i>relationship</i> antar simbol yaitu: <i>association</i> , <i>extends</i> , <i>uses</i> , <i>depends on</i> , dan <i>inheritance</i> .

2. Activity Diagram

Activity diagram memodelkan alur sebuah proses bisnis, tahapan *use case*, atau perilaku sebuah objek (*method*). Diagram ini hampir sama dengan *flowchart* yang menggambarkan urutan kerja dari sebuah *use case*.

Tabel 2.3 Tabel Simbol-Simbol Pada *Activity Diagram*

Nama Simbol	Notasi	Keterangan
<i>Initial Node</i>		Merepresentasikan awal dari sebuah proses.
<i>Action</i>		Menggambarkan sebuah tahapan/aksi.
<i>Flow</i>		Menggambarkan alur kerja.
<i>Decision</i>		Menggambarkan sebuah kondisi.
<i>Fork</i>		Menggambarkan aksi yang terjadi secara bersamaan.
<i>Activity Final</i>		Merepresentasikan akhir dari sebuah proses.



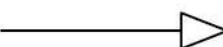
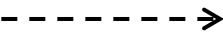

3. Class Diagram

Class diagram adalah model statis yang menggambarkan struktur dan deskripsi kelas serta hubungannya antara kelas. Kelas terdiri dari nama kelas, atribut dan operasi/metode. Atribut dan operasi (metode) dapat memiliki salah satu sifat berikut:

- a. *Private*, hanya bisa dipanggil dari dalam kelas itu sendiri. Penulisan metode/atribut diawali dengan tanda “-”.
- b. *Protected*, hanya dapat dipanggil oleh kelas yang bersangkutan dan kelas turunannya. Penulisan metode diawali dengan tanda “#”.
- c. *Public*, dapat dipanggil dari semua objek. Penulisan metode/atribut diawali dengan tanda “+”.

Tabel berikut ini menjelaskan tentang simbol hubungan antar kelas yang digunakan pada *class diagram*.

Tabel 2.4 Tabel Simbol-Simbol Pada *Class Diagram*

Nama Simbol	Notasi	Keterangan
Asosiasi / <i>Association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi Berarah / <i>Directed Association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi ini biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi		Relasi antar kelas dengan makna generalisasi – spesialisasi (umum – khusus) atau untuk menyatakan hubungan <i>inheritance</i> .
Kebergantungan / <i>Dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / <i>Aggregation</i>		Relasi antar kelas dengan makna semua-bagian (whole-part).

Bab III

Metodologi Penelitian

Dalam melaksanakan penelitian ini, Penulis membuat kerangka penelitian sebagai panduan dalam melakukan kegiatan secara berurutan mulai dari awal penelitian ini dijalankan hingga akhir hasil penelitian. Kerangka penelitian dibuat berdasarkan pengembangan metode air terjun (*waterfall*). Alasan penulis memilih metode ini adalah karena metode ini merupakan metode pengembangan tradisional yang umum digunakan dalam pembangunan perangkat lunak. Namun, metode ini tetap membuat kualitas perangkat lunak tetap terjaga karena pengembangannya yang terstruktur dan terawasi. Di sisi lain, model ini merupakan jenis model yang bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah (Binanto, 2014).

Metode *waterfall* yang digunakan dalam penelitian ini adalah metode *waterfall* berdasarkan Sommerville (2011) yang terdiri atas tahap identifikasi masalah, tahap analisis kebutuhan (*requirement analysis*), tahap desain sistem (*system design*), tahap implementasi (*implementation*), tahap pengujian (*testing*), dan tahap pemeliharaan (*maintenance*). Pada bab ini akan dibahas tentang tahap analisis kebutuhan dan desain atau perancangan sistem. Sedangkan, tahap implementasi, pengujian dan pemeliharaan akan dibahas di bab berikutnya.

3.1 Tahap Identifikasi Masalah

Dalam mengidentifikasi permasalahan yang ada, penulis melakukan observasi terlebih dahulu terhadap permasalahan yang ada. Setelah itu, penulis melakukan wawancara kepada narasumber untuk mengetahui permasalahan data yang ada di PT XYZ (terlampir naskah wawancara antara penulis dengan narasumber). Pada tahap ini diperoleh beberapa hal berikut ini.

3.1.1 Prosedur Yang Sedang Berjalan

Saat ini, proses *data cleaning* data konsumen yang dilakukan di Divisi *Consumer Care* PT XYZ masih menggunakan cara manual, yang terdiri dari tiga aktivitas utama berikut.

1. Mengecek *record* yang memiliki kemiripan.

Aktivitas ini merupakan aktivitas utama dari proses *cleaning* yang dilakukan. Langkah ini dilakukan dengan cara mengurutkan dan mengelompokkan data menggunakan beberapa atribut dengan memakai fitur *pivot* yang ada di Ms. Excel. Atribut yang digunakan untuk melakukan pembersihan data adalah *Area*, *Outlet Type*, *Name*, dan *Address*. Kemudian, data yang telah terurut dan terbagi menjadi beberapa kelompok dibaca dengan *read-scanning* untuk dapat menemukan data yang memiliki kemiripan. Jika menemukan data yang mirip, maka tidak akan langsung digabung menjadi satu *record*. Tetapi, akan dibentuk satu buah ID baru yang disebut dengan *Clean Code*.

2. Mengecek kolom yang kosong.

Memberikan tanda pada kolom yang kosong karena tidak lengkap ketika proses *input*. Terutama, terhadap atribut yang wajib untuk diisi. Kemudian, data yang kosong ini akan ditanyakan kepada pihak yang bekerja di lapangan (*field force*) untuk membantu melengkapi data yang kosong tersebut.

3. Memformat beberapa atribut yang belum sesuai dengan standar penulisan.

Merapikan struktur penulisan data yang masih tidak rapi dan tidak sesuai dengan standar penulisan. Format penulisan yang dirapikan adalah format penulisan yang ada pada kolom *Phone* dan *Fax*.

3.1.2 Master Data Konsumen Divisi *Consumer Care* di PT XYZ

Master data konsumen yang dibahas dalam studi kasus ini merupakan master data konsumen yang terdapat pada divisi *Consumer Care* yang ada pada PT XYZ. Master data yang terdapat pada divisi tersebut terdiri dari data yang besar, yaitu sekitar 25.000 data yang masih menggunakan *Ms. Excel 2010* dalam menjalankan proses operasionalnya.

Data konsumen yang dimiliki oleh divisi *Consumer Care* berasal dari data yang dikumpulkan oleh pihak distributor. Sementara, data yang berasal dari pihak distributor tersebut memiliki beberapa kesalahan yang harus dilakukan pengecekan data oleh pihak PT XYZ. Diantaranya adalah adanya *record* yang kembar.

Saat ini, pembuatan *database* untuk mengatur master data konsumen PT XYZ masih dalam proses pengembangan. Terdapat beberapa atribut yang ada pada master data konsumen tersebut. Berikut ini rincian dekripsi dari tiap atribut yang dimiliki oleh master data konsumen divisi *Consumer Care* PT XYZ di mana penulisan kamus data di bawah ini mengikuti acuan yang ada pada Schacherer (2012).

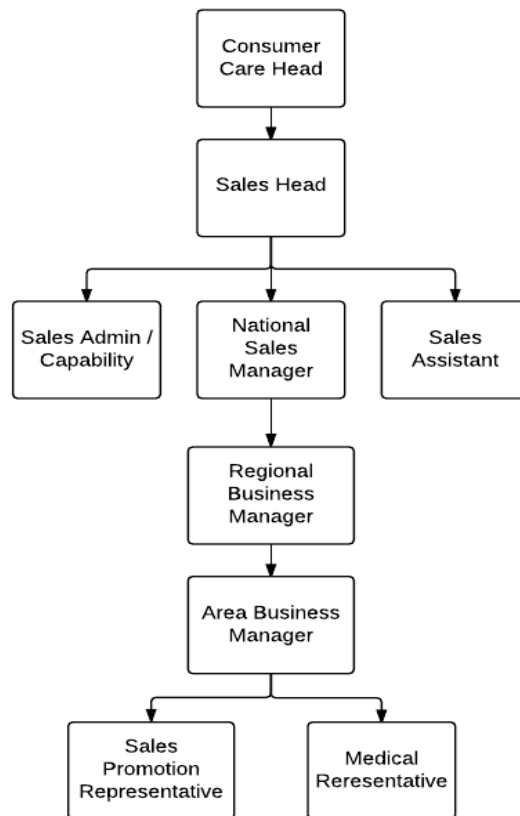
Tabel 3.1 Kamus Data Konsumen Divisi *Consumer Care* PT XYZ

Variabel	Deskripsi	Tipe Data	Sifat Pengisian	Nilai yang diharapkan
ID_Organization	Nomor ID yang dimiliki oleh toko.	<i>String</i>	Wajib	Terdiri atas angka yang unik
Name	Nama toko / <i>outlet</i>	<i>String</i>	Wajib	Penulisan nama toko yang lengkap
Address	Alamat took	<i>String</i>	Wajib	Penulisan alamat toko yang lengkap
Type	Klasifikasi yang lebih khusus dari tipe channel took	<i>String</i>	Wajib	Pilihan kategori tipe toko yang terdiri atas <i>Pharmacy, drug store, minimarket, supermarket</i> , dan lain-lain.

Variabel	Deskripsi	Tipe Data	Sifat Pengisian	Nilai yang diharapkan
Region	Pembagian wilayah regional terhadap wilayah pemasaran PT XYZ.	<i>String</i>	Wajib	Terdiri dari Sumatera, Jabotaponsa (Jakarta, Bogor, Tangerang, Pontianak, Samarinda), <i>Central</i> , dan <i>East</i>
Area	Area pembagian area pemasaran produk yang ada di PT XYZ	<i>String</i>	Wajib	Terdapat beberapa pilihan pembagian area, seperti: Bandung, Jakarta, Tangerang, dan lain-lain.
Province	Provinsi dimana lokasi toko berada	<i>String</i>	Wajib	Provinsi tempat toko berada. Terdapat beberapa pilihan provinsi yang ada di seluruh Indonesia.
City	Kota dimana lokasi toko berada	<i>String</i>	Wajib	Kota tempat toko berada. Terdapat beberapa pilihan kota yang ada di seluruh Indonesia.
Zipcode	Kode pos dimana lokasi toko berada	<i>String</i>	Wajib	Kodepos tempat toko berada
Class	Tipe kelas dari toko.	<i>String</i>	Wajib	Tipe kelas yang terdiri dari kelas A, B, dan C.
Phone	Nomor <i>handphone</i> yang dapat dihubungi	<i>String</i>	<i>Optional</i>	08xx-xxxx-xxxx
Fax	Nomor <i>fax</i> yang dapat dihubungi	<i>String</i>	<i>Optional</i>	xxxx-xxxx
Email	Email aktif yang dimiliki	<i>String</i>	<i>Optional</i>	email@provider.domain
Website	<i>Website</i> yang dimiliki took	<i>String</i>	<i>Optional</i>	www.websitename.domain
Status	Status toko apakah aktif atau tidak aktif	<i>String</i>	Wajib	<i>Active/Deactive</i>

3.1.3 Struktur Organisasi

Berikut ini merupakan struktur organisasi dari Divisi *Consumer Care* di PT XYZ beserta peran dan deskripsi kerja yang dijelaskan pada tabel berikut ini.



Gambar 3.1 Struktur Organisasi Divisi *Consumer Care* PT XYZ

Tabel 3.2 Tabel *Role* dan Deskripsi Kerja Divisi *Consumer Care* PT XYZ

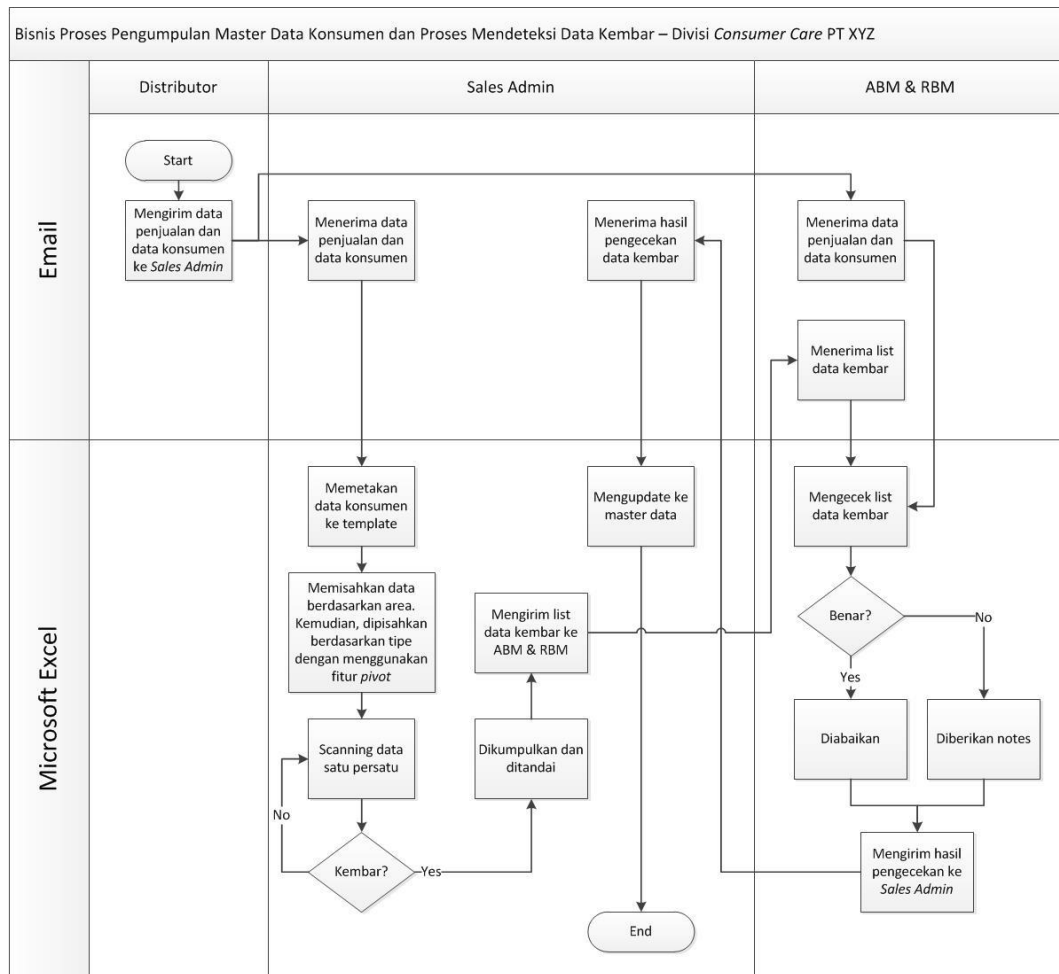
No.	Role	Deskripsi Kerja
1	<i>Consumer Care Manager</i>	<ul style="list-style-type: none"> - Mengatur dan mengkoordinasi <i>resource medical</i> dan <i>non-medical</i>, fasilitas, dan layanan. - Meningkatkan dan memperluas pasar produk <i>consumer care</i>. - Memastikan layanan yang ada telah memenuhi standar nasional dan internasional. - Mengatur registrasi/notifikasi produk untuk memastikan peningkatan bisnis di Indonesia. <p>Mengatur <i>budget</i>.</p>
2	<i>Sales Manager</i>	<ul style="list-style-type: none"> - Mengatur penjualan produk dan layanan <i>consumer care</i>. - Memastikan nilai yang konsisten dalam

No.	Role	Deskripsi Kerja
		peningkatan pendapatan penjualan. - Menempatkan dan mengatur personil-personil <i>sales</i> . - Mengidentifikasi tujuan, strategi, misi untuk meningkatkan <i>short</i> dan <i>long-term</i> penjualan dan pendapatan.
3	<i>Sales Admin / Capability</i>	- Merencanakan kebutuhan <i>sales training</i> , dan menyusun rencana <i>training</i> tahunan. - Menyusun persiapan <i>training</i> untuk perusahaan dan <i>sales force</i> distributor dalam membahas strategi dan proses menjual, manajemen penjualan, manajemen distribusi. - Mengkoordinasi penjualan dengan manajer <i>regional</i> dan <i>area</i> . - Melaksanakan kerja lapangan dan secara teratur melakukan audit untuk memastikan pelaksanaan dan implementasi di lapangan. - Menindaklanjuti dan bertanggung jawab terhadap hasil penjualan dan presentase <i>sales achievement</i> .
4	<i>National Sales Manager</i>	- Memastikan tujuan <i>sales</i> telah sesuai, tidak hanya pada region tertentu, tetapi seluruh <i>region</i> . - Mengidentifikasi kelemahan pada rencana <i>marketing</i> dan membuat sebuah aturan jika diperlukan. - Memperkirakan penjualan dalam seminggu, sebulan, atau quarter. - Menganalisa data penjualan untuk mengidentifikasi kekuatan dan kelemahan kegiatan promosi tertentu. - Mengawasi <i>budget sales force</i> perusahaan. - Menjaring dengan konsumen yang potensial dan partner bisns untuk mempromosikan produk tertentu. - Menyetujui kontrak besar.
5	<i>Sales Assistant</i>	- Membantu <i>sales manager</i> dalam mengelola program khusus dan kebutuhan operasional bisns. - Memberikan dukungan yang cepat kepada tim <i>sales</i> . - Mengatur dan mengurus administrasi yang dibutuhkan oleh tim <i>sales</i> . - Membantu dalam upaya merekrut untuk semua posisi tim <i>sales</i> yang dibutuhkan. - Mengambil peran aktif dalam melaksanakan pelatihan dan pengembangan tim.
6	<i>Regional Business Manager</i>	- Bertanggung jawab dan memantau aktivitas harian bisns regional. - Mengatur persiapan <i>viable bids</i> , proposal, dan strategi baru pada region yang dipimpin. - Melaporkan ke bagian komisaris, bekerja dengan rekan agen/distributor dan manajer Area. - Berkontribusi dalam melakukan penelitian pasar, penerapan model layanan, pengembangan model layanan, dan keberhasilan dalam penawaran produk di wilayah regional tertentu.

No.	Role	Deskripsi Kerja
7	<i>Area Business Manager</i>	<ul style="list-style-type: none"> - Bertanggung jawab dan memantau aktivitas harian bisnis area. - Melaporkan ke bagian regional manajer, bekerja dengan rekan agen/distributor dan <i>sales promotion representative</i>. - Berkontribusi dalam melakukan penelitian pasar, penerapan model layanan, pengembangan model layanan, dan keberhasilan dalam penawaran produk di wilayah area tertentu. - Memantau distributor dan <i>sales force</i> produk distributor. - Meningkatkan target dan meningkatkan profitabilitas distributor. - Mendata data seluruh konsumen yang berada di area manajemennya.
8	<i>Sales Promotion Representative</i>	<ul style="list-style-type: none"> - Membangun dan menjaga hubungan bisnis dengan konsumen pada wilayah tertentu. - Melakukan kunjungan dan presentasi ke konsumen. - Menangani masalah dan complain dari konsumen. - Menganalisa potensi pasar dan menentukan nilai dan prospektif konsumen terhadap organisasi. - Mengidentifikasi kelebihan dan membandingkan produk/layanan yang diberikan. - Mengkoordinasi <i>sales effort</i> dengan tim <i>marketing</i>, <i>sales management</i>, dan <i>accounting</i>.
9	<i>Medical Representative</i>	<ul style="list-style-type: none"> - Mengatur pertemuan dengan dokter, apoteker, dan tim medis rumah sakit. - Mengadakan presentasi ke dokter, staf atau perawat di rumah sakit dan/atau dokter dan apoteker di sektor retail. - Membangun dan mengatur hubungan positif dengan staf <i>medical</i> dan administrasi. - Mendata data seluruh dokter, apoteker, dan tim medis. - Memantau informasi tentang kegiatan pelayanan kesehatan pada area tertentu.

3.1.4 Bisnis Proses

Proses bisnis yang akan dipaparkan di sini adalah proses bisnis dalam mengumpulkan data konsumen dari pihak *Area Business Manager* kepada tim *Sales Admin / Capability* dan proses pendeteksian duplikasi data yang dilakukan secara manual. Berikut ini gambaran bisnis proses terkait proses tersebut.



Gambar 3.2 Gambar Bisnis Proses Deteksi Data Kembar Pada Master Data Konsumen Divisi *Consumer Care* PT XYZ

3.1.5 Sistem *Data Cleaning* Yang Diajukan

Berdasarkan pembahasan di atas, sistem *data cleaning* yang akan diajukan dalam penelitian ini adalah aktivitas nomor 1 dan 3 karena menurut narasumber untuk aktivitas nomor 2 tidak membutuhkan sistem khusus untuk memecahkan masalahnya. Hal-hal yang akan dicapai dalam pembuatan sistem deteksi duplikasi data pada sistem *data cleaning* dalam penelitian ini adalah:

- Proses *pra-cleaning*, yaitu proses pembersihan data dari kata, titel, tanda baca atau karakter tertentu sebelum memasuki tahap pendeteksian duplikasi data.

- b. Proses *cleaning*, yaitu proses utama yang terdiri atas pendeteksian duplikasi data.
- c. *Result*, yaitu hasil data yang telah bersih atau laporan atas duplikasi data yang telah ditemukan dengan memungkinkan *user* mengekspor hasil proses deteksi duplikasi data.

3.2 Tahap Analisa Kebutuhan Sistem

Analisis kebutuhan sistem dalam penelitian ini terdiri atas analisa kebutuhan non fungsional dan fungsional seperti yang akan dipaparkan pada sub-bab berikut.

3.2.1 Kebutuhan Non Fungsional Sistem

Kebutuhan non fungsional adalah tipe kebutuhan yang yang berisi properti perilaku yang dimiliki oleh sistem, seperti deskripsi dari fitur-fitur, karakteristik, dan batasan-batasan yang lain yang mendefinisikan sistem yang memuaskan (Al Fatta, 2007). Adapun kebutuhan non fungsional yang dipertimbangkan dalam pembuatan sistem *data cleaning* dapat dilihat pada lampiran 2.

3.2.2 Analisa Kebutuhan Fungsional Sistem

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Analisa kebutuhan fungsional sistem dilakukan untuk menganalisis apa saja kebutuhan yang diajukan untuk sistem *data cleaning* pada penelitian ini. Adapun kebutuhan fungsional mencakup deskripsi dari aktivitas-aktivitas dan layanan-layanan yang harus disediakan oleh sistem (Al Fatta, 2007). Kebutuhan fungsional yang dibutuhkan dalam penelitian ini dapat dilihat pada lampiran 2.

3.3 Perancangan Sistem

Setelah tahap analisis kebutuhan, tahap selanjutnya adalah proses perancangan sistem. Tahap perancangan sistem merupakan proses penting dalam proses perancangan aplikasi untuk menentukan hasil akhir dari rencana program

yang akan dibuat. Perancangan sebuah sistem mempengaruhi hasil akhir dari pembangunan aplikasi sehingga perlu diperhatikan proses pembuatannya. Dibutuhkan hasil analisis yang benar agar hasil dapat diimplementasikan dan sesuai dengan kebutuhan sistem. Penerapan sebuah algoritma pada salah satu fungsi dalam sistem yang akan dibangun menjadi sebuah alur penting dalam menyelesaikan kasus permasalahan yang terdapat dalam penelitian ini. Perancangan sistem terdiri atas perancangan alur algoritma, perancangan *database*, dan UML yang akan dijelaskan pada bab berikutnya.

3.4 Tahap Implementasi

Setelah melakukan perancangan sistem, tahap selanjutnya adalah implementasi. Implementasi adalah proses merubah desain menjadi bahasa pemrograman yang secara teknis biasanya dikerjakan oleh *programmer*. Pada penelitian ini, implementasi dikerjakan sendiri oleh Penulis dengan bahasa pemrograman C# dan berorientasi objek (OOP). Hasil dari tahapan ini adalah sistem *data cleaning* untuk master data konsumen Divisi *Consumer Care* PT XYZ dengan fungsi seluruh sistem yang sudah berjalan dengan baik.

3.5 Tahap Pengujian

Tahap ini merupakan tahap pengujian atas implementasi yang telah dilakukan pada tahap sebelumnya. Tahap pengujian dilakukan untuk melihat kebenaran dari logika yang dijalankan sistem dan menilai apakah implementasi yang dilakukan telah sesuai dengan yang diinginkan dalam mencapai tujuan pembuatan sistem (Pressman, 2010). Pengujian akan dilakukan dengan menggunakan metode *white box* dan *black box* serta evaluasi metode yang telah diterapkan pada sistem *data cleaning* dengan menggunakan dua sampel data (*Dlarge* data dan *Dsmall* data) berdasarkan metode yang dilakukan oleh Weis, dkk. (2008).

3.6 Jadwal Penelitian

Berikut ini merupakan rancangan waktu kerja yang Penulis rancang dalam rangka menyusun penelitian.

Tabel 3.3 Jadwal Penelitian

Jenis Kegiatan	Tahun 2015/2016																								
	April				Mei				Juni				Juli				Maret				April			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengumpulan Data																									
Studi Literatur																									
Analisis Masalah																									
Analisis Kebutuhan																									
Analisis Data																									
Perancangan Sistem																									
Seminar Proposal																									
Implementasi																									
Laporan																									
Sidang Tugas Akhir																									

Daftar Pustaka

- Al Fatta, H. (2007). *Analisis dan Perancangan Sistem Informasi Untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: Amikom.
- Azma, S. (2006). Pembuatan Alat Bantu Dalam Proses Data Cleaning Pada Intra-Governmental Access to Shared Information System (IGASIS) . Bandung, Jawa Barat, Indonesia.
- Binanto, I. (2014). *Analisa Metode Classic Life Cycle (Waterfall) Untuk Pengembangan Perangkat Lunak Multimedia*. Yogyakarta.
- Chapman, A. D. (2005). Principles and Methods of Data Cleaning - Primary Species and Species-Occurrence Data, version 1.0 . (p. 1). Queensland, Australia: Global Biodiversity Information Facility.
- Couto, P. D. (2012, October). Support for User Interaction in a Data Cleaning Process (Dissertation). Germany.
- Dan, C. (2011). *Beginning C# Object-Oriented*. New York: Apress.
- Edition, I. D. (1993). *Data Dictionary*. Retrieved April 16, 2015, from Wikipedia.org: http://en.wikipedia.org/wiki/Data_dictionary
- Friedman, C., & Sideli, R. (1992). Tolerating Spelling Errors During Patient Validation. *Computers and Biomedical Research*, (pp. 486-509). New York.
- Guo, L., Wang, W., Chen, F., Tang, X., & Wang, W. (2012). A Similar Duplicate Data Detection Method Based on Fuzzy Clustering for Topology Information. In *PRZEGLĄD ELEKTROTECHNICZNY (Electrical Review), 01b* (pp. 26-31).
- Hernandez, M. A. (1995). A Generalization of Band Joins and The Merge/Purge Problem (Thesis Proposal). New York.
- Hernandez, M. A., & Stolfo, S. J. (1995). The Merge/Purge Problem for Large Database. 128-129.
- Huda, N. M. (2010). Aplikasi Data Mining Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa. Semarang, Indonesia.

- Lee, M. L., Lu, H., Ling, T. W., & Ko, Y. T. (1999). Cleansing Data for Mining and Warehousing.
- Low, W. L., Lee, M. L., & Ling, T. W. (2001, May 20). A Knowledge-Based Approach for Duplicate Elimination in Data Cleaning. Singapore.
- Maimon, O., & Rokach, L. (2005). *The Data Mining and Knowledge Discovery Handbook*. Tel Aviv, Israel: Springer.
- Maletic, J. I., & Marcus, A. (2000). Data Cleansing: Beyond Integrity Analysis. Memphis.
- Navarro, G. (2001). A Guided Tour to Approximate String. *ACM Computing Surveys*. Santiago.
- Ningsih, V. M. (2009). *OOP vs Prosedural*. Telemetri: <http://blog.neotelemetri.com/index.php/pemrograman/8-oop-vs-prosedural>
- R, A., & Narashiman, K. (2014). A Simplified Framework for Data Cleaning and Information Retrieval in Multiple Data Source Problems. *International Journal of Innovative Research in Science, Engineering and Technology*.
- Rahaman, G. M., Rahman, A., & Ripon, K. S. (2010, December 12). A Domain-Independent Data Cleaning Algorithm for Detecting Similar-Duplicates. Bangladesh.
- Recchia, G., & Max, L. (2013). A Comparison of String Similarity Measures for Toponym Matching. *ACM SIGSPATIAL COMP'13*. New York.
- Rouf, A. (2012). Pengujian Perangkat Lunak Dengan Metode White Box dan Black Box. Semarang.
- Schacherer, C. W. (2012). SAS® Data Management Technique: Cleaning and Transforming Data for Delivery of Analytic Datasets.
- Sommerville, I. (2011). *Software Engineering 9th Edition*. San Fransisco: Addison-Wesley.
- Sulistyorini, P. (2009). Pemodelan Visual dengan Menggunakan. *Jurnal Teknologi Informasi DINAMIK*, 23-29.
- T. Sembok, T. M., & Abu Bakar, Z. (2011). Effectiveness of Stemming and N-grams String Similarity Matching on Malay Documents. *International*

Journal of Applied Mathematics and Informatics, (pp. 208-215). Bangi, Malaysia.

Tian, Z., Lu, H., Ji, W., Zhou, A., & Tian, Z. (2001). An n-gram-based Approach for Detecting Approximately Duplicate Database Records. *Springer Verlag*.

Yannakoudakis, E. J., & Angelidakis, G. (1988). An Insight into The Entropy and Redundancy of The English Dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (pp. 960-970).

Lampiran 1 – Wawancara

Berikut ini merupakan rangkuman beberapa wawancara yang penulis (selanjutnya ditulis dengan P) lakukan dengan calon *user*, yaitu Widi Rizky Ayudya (selanjutnya ditulis dengan WRA) selaku Staf Divisi *Consumer Care* di PT XYZ:

P : Apakah benar selama ini Mba Widi merasa sangat kewalahan dalam memvalidasi data *Consumer Care*?

WRA : Betul sekali. Karena data yang Saya *handle* sangat banyak dan terus bertambah setiap bulan.

P : Kalau boleh tahu berapa banyak data yang sekarang Mba Widi *handle*?

WRA : Sekitar 25.000 *row* data dan data akan terus bertambah tiap bulannya sekitar 1.000 data.

P : Dengan menggunakan *tools* atau *software* apa biasanya Mba Widi melakukan proses validasi data *Consumer Care*?

WRA : Proses validasi yang Saya lakukan menggunakan cara manual dan menggunakan *Ms. Excel* 2010 untuk melakukan validasi data.

P : Kalau boleh tahu, untuk data konsumen *Consumer Care*-nya sendiri itu disimpan di dalam sistem *database* atau disimpan dengan menggunakan *excel*?

WRA : Masih manual menggunakan *excel*.

P : Untuk sumber datanya sendiri didapat darimana ya, Mba?

WRA : Jadi, sumber data konsumen yang PT XYZ punya ini berasal dari distributor PT XYZ. Namun, jika ada data yang menurut Saya tidak *valid* atau janggal maka Saya akan bertanya ke pihak ABM (*Area Business Manager*) PT XYZ untuk menanyakan kevalidan data tersebut.

P : Pihak distributor mengirimkan data konsumen biasanya

menggunakan *software* atau *tools* seperti apa dan *file* datanya itu berbentuk *excel* juga kah?

WRA : Biasanya dikirim melalui *email* dan benar sekali dikirimnya berupa *file .xls*.

P : Apa saja sih *problem* dari proses validasi data yang Mba Widi lakukan?

WRA : Problemnya itu ada data yang kembar. Seperti ini.. (memberikan contoh datanya). Kemudian, terkadang kolom *outlet category* tidak sesuai dengan deskripsi namanya. (memberikan contoh datanya) Nah, kalau kasus seperti itu, penyelesaiannya adalah ditanyakan ke pihak ABM. Selain itu, seperti kolom-kolom yang kosong. Kalau ada kolom yang wajib diisi kemudian kosong, maka Saya akan tanyakan lagi ke pihak ABM. Setelah itu, seperti penulisan-penulisan yang belum rapi. Contohnya, kolom *phone* dan *fax* ini. Ini Saya rapikan penulisannya. Karena *urgency*-nya tidak sepeenting yang lain, jadi untuk kasus ini tidak Saya dahulukan dan memang sekali dua kali saja Saya kerjakan, jika sempat.

P : Lalu, bagaimana cara Mba Widi menemukan data kembar dari jumlah data yang sangat besar ini?

WRA : Caranya, pertama data ini Saya urutkan terlebih dahulu. Jadi, Saya menggunakan fitur *pivot* untuk mengurutkan datanya. Begini.. (memeragakan caranya). Pertama, Saya blok terlebih dahulu. Kemudian saya *insert – Pivot*. Setelah itu, Saya *drag Area* ke bagian *Row Labels*. Nah, ini data tiap Area akan secara otomatis berkelompok berdasarkan area masing-masing. Kemudian, Saya *drag Outlet Type* ke *Row Labels*. Kemudian, data yang sudah terbagi ke tiap area akan terbagi lagi berdasarkan tipe *outlet* masing-masing. Setelah itu, Saya *drag Name* dan *Address* ke *Row Labels*. Lalu, *drag Name* ke *Values*. Saya *drag* ke *Values* agar Saya bisa tahu jumlah dari tiap *Row* yang sudah diurutkan. Karena kalau jumlahnya lebih dari satu, bisa jadi data itu adalah data kembar. Nah, Setelah semua

terurut, Saya baca satu-satu. Saya *scanning*. Seperti ini... Nanti kalau ada yang datanya terlihat agak mirip. Maka, akan saya cek keseluruhan datanya apakah benar-benar mirip atau tidak. Karena bisa jadi, seperti toko-toko seperti Kimia Farma, *Carrefour*, pokoknya toko-toko yang tersebar dimana-mana. Itu biasanya, alamatnya sama karena mereka melakukan pembelanjannya secara terpusat. Namun, sebenarnya *outlet* tersebut berbeda letaknya. Kalau memang menemukan kasus yang seperti itu, artinya data tersebut tidak kembar.

P : Bukankah untuk kolom *Outlet Type* datanya terkadang tidak benar seperti yang tadi Mba Widi katakan?

WRA : Nah, itu dia masalahnya. Coba bayangkan kalau misalnya Saya tidak membagi datanya berdasarkan *Outlet Type*-nya. Dalam satu area Saya harus mengecek sekitar 1.000 data secara bersamaan. Tidak mungkin bukan? Makanya, Saya bagi saja menjadi *Outlet Type* yang berbeda. Ya, walaupun tidak semua akurat. Tapi, nilai ketidakakuratan *Outlet Type* itu hanya sekitar 10% lah. Hehe. Jadi, setidaknya kalau dengan membagi datanya ke tipe *outlet*-nya masing-masing akan sangat membantu Saya untuk menemukan data kembar.

P : Mengapa tidak menggunakan kolom *City* atau *Subcity* untuk memisahkan datanya Mba Widi?

WRA : Karena data yang ada di kolom *City* dan *Subcity* tidak sepenuhnya benar. Sedangkan, kalau kolom *Area*, Karena memang area itu merupakan pembagian area *marketing* dari tiap ABM PT XYZ.

P : Bukankah bisa dikatakan apa yang Mba Widi lakukan ini tidak akan ada habisnya karena data akan terus bertambah bukan, Mba?

WRA : Bisa dikatakan begitu. Tapi, setidaknya, Saya disini yang akan *maintain* data. Coba bayangkan kalau tidak ada yang *maintain* datanya.

P : Kalau misalnya ada sistem yang melakukan apa yang Mba Widi kerjakan secara otomatis. Apakah Mba Widi setuju?

WRA : Wah.. setuju sekali. Karena jujur itu memudahkan pekerjaan Saya sekali. Kadang, untuk *maintain* data ini suka tidak bisa semua saya *handle*. Jadi, kalau misalkan ada sistem yang bisa secara otomatis membaca data kembar saja itu sudah memudahkan pekerjaan Saya.

Lampiran 2 – *Requirement Elicitation*

Requirement Elicitation

Requirement Elicitation Sistem Data Cleaning Divisi Consumer Care PT XYZ

Requirement Elicitation Tahap 1

Fungsional	
No.	Analisa Kebutuhan
Saya ingin sistem dapat	
1.	Mengijinkan <i>user</i> dapat melakukan <i>login</i> ke dalam sistem.
2.	Mengijinkan <i>user</i> dapat memasukan atau mengimpor data konsumen Divisi <i>Consumer Care</i> ke dalam <i>database</i> .
3.	Mengijinkan <i>user</i> untuk melakukan pendeteksian duplikasi data.
4.	Mengijinkan <i>user</i> untuk merapikan format penulisan <i>phone</i> dan <i>fax</i> .
5.	Mengijinkan <i>user</i> untuk menampilkan hasil deteksi duplikasi data.
6.	Mengijinkan <i>user</i> untuk mengekspor atau mengunduh hasil pendeteksian duplikasi data ke dalam <i>file excel</i> .
7.	Mengijinkan <i>user</i> untuk menyimpan hasil perubahan penulisan format <i>phone</i> dan <i>fax</i> ke dalam <i>database</i> .

Non-fungsional	
No.	Analisa Kebutuhan
Saya ingin sistem dapat	
1.	Memiliki hasil pendeteksian duplikasi data yang cukup akurat.
2.	Mampu berjalan dengan berbasis web.
3.	Menampilkan tampilan <i>web</i> yang sesuai dengan <i>template</i> sistem yang ada pada PT XYZ.

Requirement Elicitation Tahap 2

Elisitasi Tahap II dibentuk berdasarkan Elisitasi Tahap I yang diklasifikasikan melalui metode MDI (*Mandatory, Desirable, Inessential*). Berikut penjelasan dari beberapa *requirement* yang mendapatkan opsi M, D, atau I.

Fungsional				
No.	Analisis Kebutuhan	M	D	I
Saya ingin sistem dapat				
1.	Mengijinkan <i>user</i> dapat melakukan <i>login</i> ke dalam sistem.	√		
2.	Mengijinkan <i>user</i> dapat memasukan atau mengimpor data konsumen Divisi <i>Consumer Care</i> ke dalam <i>database</i> .	√		
3.	Mengijinkan <i>user</i> untuk mereset atau menghapus data konsumen yang ada di dalam <i>database</i> .	√		
4.	Mengijinkan <i>user</i> untuk melakukan pendeteksian duplikasi data.	√		
5.	Mengijinkan <i>user</i> untuk merapikan format penulisan <i>phone</i> dan <i>fax</i> .		√	
6.	Mengijinkan <i>user</i> untuk menampilkan hasil deteksi duplikasi data dan perubahan penulisan format <i>phone</i> dan <i>fax</i> .	√		
7.	Mengijinkan <i>user</i> untuk mengekspor atau mengunduh hasil pendeteksian duplikasi data ke dalam <i>file excel</i> .	√		
8.	Mengijinkan <i>user</i> untuk menyimpan hasil perubahan penulisan format <i>phone</i> dan <i>fax</i> ke dalam <i>database</i> .		√	

Non Fungsional				
No.	Analisis Kebutuhan	M	D	I
Saya ingin sistem dapat				
1.	Memiliki hasil pendeteksian duplikasi data yang cukup akurat.	√		
2.	Mampu berjalan dengan berbasis web.	√		
3.	Menampilkan tampilan <i>web</i> yang sesuai dengan <i>template</i> sistem yang ada pada PT XYZ.	√		

Keterangan:

M = Mandatory (yang diinginkan),

D = Desirable (diperlukan),

I = Inessential (yang tidak diinginkan)

Requirement Elicitation Tahap 3

Berdasarkan Elisitasi Tahap II di atas, dibentuklah Elisitasi Tahap III yang diklasifikasikan kembali dengan menggunakan metode TOE (*Technical, Operational, Economic*) dengan opsi LMH (*Low, Medical, High*). Berikut adalah *requirement elicitation* yang ada pada tahap 3.

Fungsional										
Feasibility		T			O			E		
		L	M	H	L	M	H	L	M	H
No.	Analisis Kebutuhan									
Saya ingin sistem dapat										
1.	Mengijinkan <i>User</i> dapat melakukan <i>login</i> ke dalam sistem.		√				√	√		
2.	Mengijinkan <i>User</i> dapat memasukan atau mengimpor data konsumen Divisi <i>Consumer Care</i> ke dalam <i>database</i> .			√			√	√		
3.	Mengijinkan <i>User</i> untuk mereset atau menghapus data konsumen yang ada di dalam <i>database</i> .	√				√		√		
4.	Mengijinkan <i>User</i> untuk melakukan pendeteksian duplikasi data.			√			√		√	
5.	Mengijinkan <i>User</i> untuk merapikan format penulisan <i>phone</i> dan <i>fax</i> .			√		√			√	
6.	Mengijinkan <i>User</i> untuk menampilkan hasil deteksi duplikasi data dan perubahan penulisan format <i>phone</i> dan <i>fax</i> .	√					√	√		
7.	Mengijinkan <i>User</i> untuk mengekspor atau mengunduh hasil pendeteksian duplikasi data ke dalam <i>file excel</i> .			√			√		√	

8.	Mengijinkan <i>User</i> untuk menyimpan hasil perubahan penulisan format <i>phone</i> dan <i>fax</i> ke dalam <i>database</i> .		√			√		√		
----	---	--	---	--	--	---	--	---	--	--

Non Fungsional										
Feasibility		T			O			E		
		L	M	H	L	M	H	L	M	H
No.	Analisis Kebutuhan									
Saya ingin sistem dapat										
1.	Memiliki hasil pendeteksian duplikasi data yang cukup akurat.			√			√			√
2.	Mampu berjalan dengan berbasis web.		√				√	√		
3.	Menampilkan tampilan <i>web</i> yang sesuai dengan <i>template</i> sistem yang ada pada PT XYZ.		√				√	√		

Keterangan:

T = *Technical* O = *Operational* E = *Economic*

M = *Middle* L = *Low* H = *High*