

DETEKSI KESALAHAN TRANSFORMASI DATA CUACA

Wiwin Suwarningsih

Pusat Penelitian Informatika LIPI,
Komplek LIPI Gd.20 Lt.3 Jl. Cisitu 21/154-d Sangkuriang Bandung
wiwin.suwarningsih@lipi.go.id

Abstrak

Sebuah sistem online monitoring cuaca dan sistem peringatan dini banjir membutuhkan suatu data masukan yang benar, karena bila data cuaca yang diolah salah maka akan menghasilkan sebuah informasi yang salah pula. Proses transfer data dari alat sensor ke stasiun pencatat berupa data text yang kemudian diterjemahkan kedalam sintaks SQL. Pada tahap transformasi dari bentuk text ke sintaks SQL dibutuhkan suatu pendeteksi kesalahan data agar data yang tersimpan di basisdata bila akan digunakan untuk aplikasi dapat menghasilkan sebuah informasi yang benar. Metoda yang digunakan untuk pendeteksian kesalahan transformasi data cuaca ini adalah pemodelan referensi, metoda ini membahas persyaratan dan pengembangan pemodelan referensi yang mencerminkan kebutuhan untuk konfigurasi data atau informasi. Pemetaan dan analisa masalah pada metoda ini menggunakan notasi EPC (Event-driven Process Chain), sehingga pemodelan konfigurasi referensi dan prinsip-prinsip desain yang sesuai dapat digunakan dalam skenario pendeteksian kesalahan pada saat transformasi data cuaca dari bentuk text menjadi sintaks SQL. Hasil akhir dari penelitian ini adalah akurasi data cuaca tetap terjaga dengan didukung proses pendeteksian yang relatif cepat

Kata kunci :

Transformasi data, Pemodelan Referensi, Cuaca, Konfigurasi Data, EPC

Abstract

An online weather monitoring system and flood early warning system requires correct data input, because if input data that is processed incorrectly it will produce a false information as well. The data is transferred from sensors to the recording station in the form of text data and then translated into SQL syntax. At the transformation stage from text into SQL syntax, it needs data error detection in order the

data that stored in database will produce the correct information whenever it retrieve by the application.

Method that will be used in error detection of weather data transformation is reference model, which addresses the requirements and development of reference model that reflects the need for configuration of data or information. Mapping and analysis of the problem is done using the notation EPC (Event-driven Process Chain), so that the reference configuration modeling and design principles that fit can be used in error detection scenarios during the transformation of weather data from text into the SQL syntax. The end result of this research is the accuracy of weather data is maintained with the support of rapid detection process.

Keywords: Data transformation, reference model, weather, data configuration, EPC.

I. PENDAHULUAN

Mengelola data cuaca membutuhkan suatu teknik yang benar dan optimal, karena data cuaca merupakan jenis data yang krusial. Mengapa dikatakan krusial ? karena dengan data cuaca ini kita dapat menggunakan untuk prediksi kondisi cuaca suatu daerah apakah dalam kondisi panas, hujan atau berawan. Data cuaca dapat digunakan pula untuk memprediksi kondisi bencana seperti banjir, tanah longsor ataupun kekeringan. Di sektor pertanian data cuaca dapat digunakan untuk memprediksi musim tanam dan musim panen. Banyak sekali informasi yang dapat dihasilkan dengan menggunakan data cuaca ini.

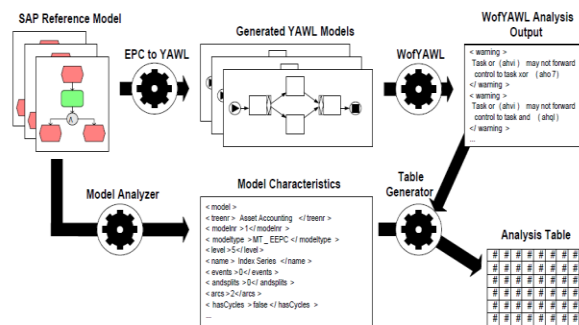
Proses perekaman data yang dilakukan saat ini adalah menggunakan sistem *data logger* yang dipasang di stasiun cuaca. Kemudian data tersebut dikirim dengan menggunakan sistem jaringan nirkawat dengan durasi waktu setiap 10 menit sekali. Di sisi stasiun pencatat data tersebut direkam, proses

perekaman data ini dilakukan dengan cara merubah bentuk data text menjadi sintkas SQL. Perubahan data dari bentuk text menjadi sintkas SQL dilakukan untuk memudahkan penyimpanan dan pengelolaan basisdata cuaca.

Berdasarkan hal tersebut maka diperlukan suatu proses pendeteksian kesalahan transformasi atau perubahan data dari bentuk text menjadi sintkas SQL, ini dilakukan agar data yang tersimpan di basisdata cuaca merupakan data yang benar agar apabila data cuaca tersebut dibutuhkan sebagai data masukan untuk aplikasi sistem peringatan dini banjir akan menghasilkan informasi yang benar.

II. PENELITIAN TERKAIT

Model referensi adalah model konseptual generik yang direkomendasikan untuk domain tertentu (Fettke and Loss). Dengan metoda ini analisa dan pemodelan masalah dapat digambarkan berdasarkan fungsi dan kejadian setiap obyek yang terlibat dalam permasalahan tersebut. (Fettke and Loss, 2003) (Dumas and Hofstede, 2005). Penerapan model referensi dimotivasi oleh paradigma 'design-for/by-reuse', yaitu mempercepat proses pemodelan dengan menyediakan repositori proses yang relevan dan struktur idealnya dengan menggunakan modus 'plug & play' modus(Dumas and Hofstede, 2005) (Mendling,2008). Dengan demikian, pemodelan referensi terkait erat dengan penggunaan kembali model informasi dengan menyediakan solusi model generik yang dapat disesuaikan dengan model tertentu yang mencerminkan kebutuhan individu. Gambaran umum desain analisis model referensi dapat dilihat pada gambar 1.



Gambar.1. Gambaran Umum Desain Analisis(Mendling, 2008)(Rosemann, 2007)

Pada gambar 1 dapat dilihat sebagai masukan untuk langkah-langkah analisis menggunakan sintaks XML dari model referensi. Langkah pertama EPC diubah ke bentuk YAWL dan hasil transformasi adalah sebuah file XML untuk setiap EPC dalam model referensi(Dumas and Hofstede, 2005) (Mendling, 2008)(Rosemann, 2007). Model YAWL tersebut kemudian dianalisis untuk menghasilkan penemuan kesalahan sintaks XML. Langkah selanjutnya adalah analisa model ekstrak informasi deskriptif seperti jumlah elemen dari tipe elemen tertentu dan apakah ada siklus untuk setiap model EPC. Sebuah file XML merupakan karakteristik dari model ini, kemudian digabungkan dengan data keluaran YAWL berdasarkan ID masing-masing EPC, dan ditulis ke tabel analisis dalam format HTML(Mendling, 2008)(Rosemann, 2007).

Model EPC adalah sebuah grafik yang menggambarkan peristiwa dan fungsi(Dumas and Hofstede, 2005) (Mendling, 2008)(Rosemann, 2007). Model ini menyediakan berbagai konektor yang memungkinkan eksekusi paralel alternatif dan proses. Aliran ini ditentukan oleh operator logika AND, XOR dan OR. Notasi yang digunakan pada model EPC sangat sederhana dan mudah dimengerti.

III. METODA PENELITIAN

Metoda yang digunakan pada penelitian ini adalah RML (*Reference Modeling Languages*), metoda ini membahas persyaratan dan pengembangan pemodelan referensi yang mencerminkan kebutuhan untuk konfigurasi data atau informasi. Pengembangan dan konfigurasi masalah menggunakan notasi EPC.

IV. HASIL DAN PEMBAHASAN

Penerimaan data cuaca yang dilakukan saat ini berbentuk data text (lihat gambar 2). Data dikirim setiap 10 menit dikirim dengan format tanggal, jam kemudia parameter cuaca yang terdiri dari suhu, curah hujan, kelembaban, kecepatan angin dan arah

12-11-05	10:05:14	;+024,18;+002,01;+067,78; +1,45;+106
----------	----------	---

Gambar 2. Data Text Cuaca

Proses konversi text ke bentuk SQL dilakukan setiap saat karena informasi data cuaca ini digunakan

pula untuk informasi secara online. Setelah data text diterima selanjutnya dilakukan tranformasi dari bentuk text ke dalam sintkas SQL (lihat gambar 3).

```
// baca text file
$file = "/convert/cuaca.txt";
$f = fopen($file, "r");
$data = fread($f, filesize($file));
fclose($f);

//pencarian lokasi TAB untuk karakter (";" atau " ")
$output = str_replace("\t ; \n", " ", $data);

// function explode utk memecah setiap baris
$output = explode("\n", $output);

// function foreach utk looping jumlah array dan explode
element

foreach($output as $var) {
$tmp = explode(" |", $var);
$Tanggal = $tmp[0];
$Jam = $tmp[1];

$Suhu = $tmp[2];
$CurahHujan = $tmp[2];
$Kelembaban = $tmp[2];
$KecAngin = $tmp[2];
$ArahAngin = $tmp[2];
}
```

Gambar 3. Konversi Txt ke MySQL

IV.1 Elaborasi Proses Konversi txt ke MySQL dalam Notasi EPC

Elaborasi proses konversi txt ke MySQL dengan notasi EPC ini merupakan penggambaran tahap-demi tahap analisis terhadap syntax yang digunakan untuk melihat kemungkinan terjadinya kesalahan bentuk format data txt. Pada gambar 3 proses transformasi dimulai dengan membaca file text, fungsi yang digunakan FOpen untuk membuka file text, fungsi FRead untuk membaca file text dan FClose untuk menutup file text. Dengan menggunakan kombinasi konektivitas OR_SPLIT. Jika proses pembacaan file text berhasil maka dilakukan pencarian lokasi tab pemisah dengan menggunakan fungsi Str_Replace dimana identifikasi tab pemisah berupa spasi dan titik koma (;).

Tahap selanjutnya adalah menggabungkan fungsi Str_Replace dengan fungsi Exlpode dengan

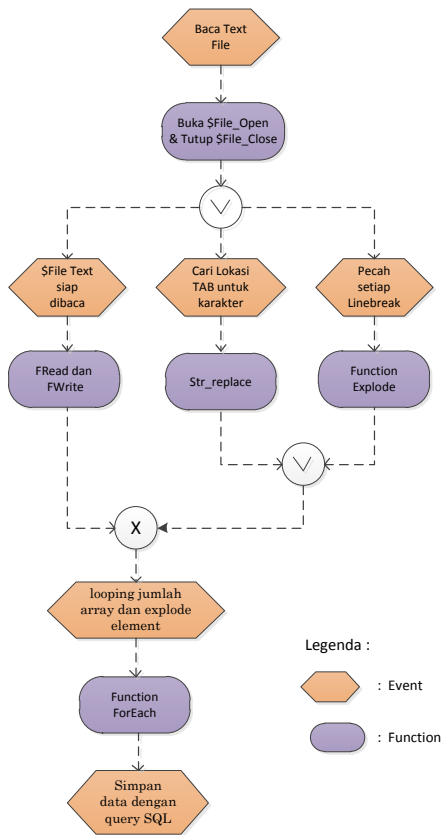
menggunakan relasi OR_JOIN untuk proses pemecahan setiap baris data. Tahap ini dilakukan untuk memisahkan data yang datang pertama dengan data yang datang berikutnya. Kemudian dilakukan konektivitas XOR_JOIN untuk alokasi ruang elemen data dan penyesuaian data dalam file text. Fungsi ForEach pada proses penyimpanan data dalam array digunakan untuk mengatur alokasi data tersebut. Apabila alokasi data sudah dilakukan maka selanjutnya adalah menyimpan data dengan perintah SQL yaitu INSERT INTO tabel Data_Cuaca. Proses memasukan data dalam tabel ini merupakan tahap akhir dari diagram EPC.

IV.2 Deteksi Kesalahan Transformasi Data dalam Notasi Petri Net.

Proses menterjemahkan dari notasi EPC ke notasi PetriNet adalah untuk membuat aliran kerja dalam bentuk jaringan Petri (Petri Net)(Mendling, 2008)(Rosemann, 2007). Hal yang menjadi pertimbangan analisis deteksi kesalahan dengan cara mengubah notasi EPC menjadi notasi PetriNet adalah adanya beberapa masalah dengan konektivitas pada notasi EPC(Dumas and Hofstede, 2005)(Mendling, 2008).

Proses pendeteksian kesalahan dapat dilakukan dengan menggunakan notasi PetriNet karena Petrinet merupakan model yang penggunaannya untuk sumber daya tertutup, sistem aktif yang non-reaktif[7]. Jika kita perhatikan untuk jenis data cuaca ini merupakan sumberdaya tertutup. Sistem pengiriman datanya pun dikategorikan sistem aktif yang no reaktif karena dibuat secara otomatis mengirimkan data setiap 10 menit sekali.

Pertimbangan lain tidak menggunakan notasi aktivitas diagram karena diagram aktivitas UML memodelkan untuk sistem terbuka dan sistem reaktif(Eshuis and Wieringa, 2003).



Gambar 4. Tranformasi Data Text ke Sintaks SQL dalam Notasi EPC

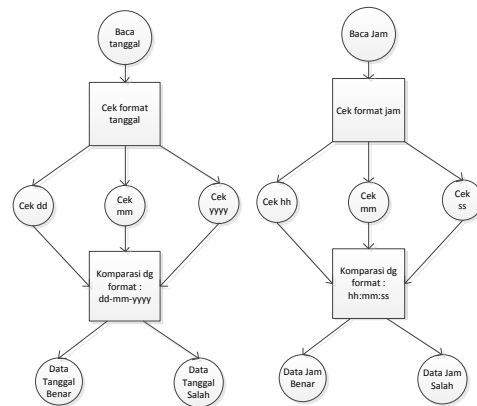
IV.2.1 Deteksi kesalahan pada OR_SPLIT untuk fungsi Str_Replace dan fungsi explode.

Jika konektivitas OR_SPLIT pada fungsi Str_Replace tidak dapat membedakan tab pemisah antar data maka proses pembentukan variabel tidak dapat didefinisikan. Pembentukan variabel ini digunakan untuk proses pengaturan ruang penyimpanan data dimana tahap ini harus menunggu proses pemisahan data sampai selesai. Hal ini dilakukan untuk menghindari kerancuan pada saat proses pemisahan data pada fungsi Str_Replace.

Koreksi kesalahan data text dilakukan pada fungsi Str_Replace. Proses mendeteksi kesalahan yang pertama adalah membandingkan format standar tanggal dan jam. Jika ada salah satu format yang salah maka dianggap data salah (*error*).

Data tanggal pada file text dibandingkan dengan format standar yang terdiri dari dd-mm-yyyy. Dimana dd adalah tanggal, menggunakan dua digit antara 1 sampai dengan 31, mm adalah bulan antara 1 s.d.12, dan yyyy adalah tahun antara 2011 s.d. 2020. Pemisah antara data tanggal, bulan dan tahun adalah tanda strip '-'. Gambaran pendeteksian kesalahan untuk format tanggal dan jam dapat dilihat pada gambar 5.

Sedangkan format standar untuk jam adalah hh:mm:ss, dimana hh adalah jam dengan rentang nilai 00-23, mm adalah menit dengan rentang nilai 00-59, dan ss adalah detik dengan rentang nilai 00-59. Pemisah antara jam, menit dan detik menggunakan titik dua ':'.



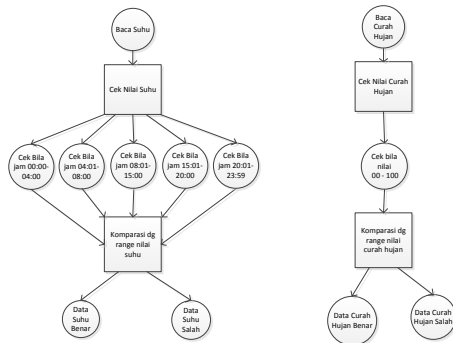
Gambar 5. PetriNet Deteksi Data Tanggal dan Data Jam

Deteksi kesalahan selanjutnya adalah pembacaan data suhu, dimana data suhu ini tergantung pada data jam. Hal ini dikarenakan rentang nilai suhu untuk kondisi pagi, siang, sore dan malam hari berbeda-beda (lihat tabel 1).

Tabel 1. Rentang Nilai Suhu berdasarkan jam (data diambil dari stasiun cuaca Di Lembang)

Jam	Suhu (° Celcius)
00:00 – 04:00	15,00 – 19,00
04:01 – 08:00	19,00 – 22,00
08:01 – 15:00	22,00 – 29,00
15:01 – 20:00	22,00 – 25,00
20:00 – 23:59	19,00 – 22,00

Berdasarkan tabel 1, bila data suhu yang dibaca tidak sesuai dengan rentang nilai tersebut maka dianggap data suhu salah.



Gambar 6. PetriNet Deteksi Data Suhu dan Curah Hujan

Pada gambar 6 untuk mendeteksi nilai curah hujan hanya menggunakan satu rentang nilai saja yaitu antara 0,00 sampai dengan 100,00. Begitu juga dengan kecepatan angin rentang nilainya antara 0,00 sampai dengan 10,00. Sedangkan untuk parameter kelembaban nilai datanya dibandingkan berdasarkan kondisi jam (lihat tabel.2), sehingga proses pendeteksiannya hampir sama dengan suhu

Tabel 2. Rentang Nilai Kelembaban berdasarkan jam (data diambil dari stasiun cuaca Di Lembang)

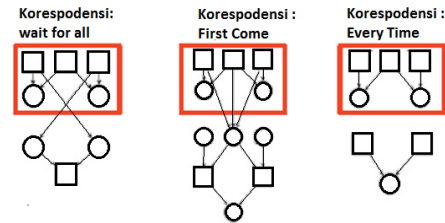
Jam	Kelembaban
00:00 – 08:00	94 – 98
08:01 – 15:00	70 – 94
15:01 – 23:59	94 – 98

Parameter terakhir yang diuji kebenaran nilainya adalah arah angin, disini kami menggunakan skala 00-150° LU (lintang Utara). Bila data arah angin berada dalam rentang nilai tersebut maka data arah angin dianggap benar, jika diluar rentang nilai tersebut maka dianggap bahwa data arah angin salah.

IV.2.2 Deteksi kesalahan pada XOR_JOIN untuk looping jumlah array dan explode elemen.

Deteksi kesalahan dengan menggunakan konektivitas XOR_JOIN pada proses pengulangan data larik (array) dan fungsi explode adalah untuk menentukan apakah data text dalam satu baris benar atau tidak. Karena jika ada salah satu data salah maka data tersebut tidak perlu ditransformasi ke sintaks SQL dan data akan dihapus. Akan tetapi jika data benar maka data akan ditransformasi ke sintaks SQL

dan disimpan dalam basisdata. Proses analisa kesalahan ini menggunakan fungsi 'wait-for-all', 'first-come' atau 'every-time'(lihat gambar 7) (Mending, 2008)(Rosemann,2007)(Recker, 2006).



Gambar 7. Fungsi Wait-For-All, First-Come dan Every-Time untuk XOR_JOIN

Karakteristik dari ketiga fungsi tersebut secara umum adalah mengintepretasikan setiap kejadian kedalam suatu aksi yang mengarah pada penempatan tahapan proses selanjutnya. Perbandingan cara kerja tiga fungsi 'wait-for-all', 'first-come' atau 'every-time' dapat dilihat pada tabel 3.

Tabel 3. Perbandingan fungsi 'wait-for-all', 'first-come' atau 'every-time'(Aals, dkk, 2006)(Eshuis and Wieringa, 2003)

Fungsi	Cara Kerja
<i>wait-for-all</i>	Pendeteksian dilakukan dengan cara menunggu fungsi yang telah diaktifkan selesai secara keseluruhan. Kelebihan : - Sesuai untuk jenis data yang saling berkaitan - Kelemahan : - Proses deteksi lama. - Tidak sesuai untuk data dengan jumlah yang besar
<i>first-come</i>	Pendeteksian dilakukan dengan cara menunggu sampai ada satu fungsi yang selesai, fungsi yang lain diabaikan. Kelebihan : - Proses deteksi lebih cepat. - Sesuai untuk data dengan jumlah besar Kelemahan : Terjadi penumpukan fungsi yang telah diaktifkan.
<i>every-time</i>	Pendeteksian dapat dilakukan beberapa kali. Kelebihan : Proses deteksi dapat dilakukan lebih dari satu untuk satu jenis data Kelemahan : - Memicu munculnya ambigu - Tidak sesuai untuk jumlah data yang besar

Berdasarkan gambar 7 proses pembacaan data dan deteksi kesalahan sebagai berikut :

- Fungsi *Wait-For-All* untuk data text dalam satu baris akan dibaca sampai akhir data, sehingga bila seluruh data benar maka data text akan ditransformasi ke sintaks SQL.
- Fungsi *First-Come* akan melakukan pembacaan data dari awal, ketika pada data pertama atau kedua atau ketiga salah maka proses akan berhenti dan data dianggap sebagai data yang salah dan tidak akan dilakukan transformasi ke sintaks SQL.
- Fungsi *Every-Time* untuk jenis data cuaca ini tidak sesuai karena proses pembacaan data menggunakan larik yang dibaca dari awal baris sampai dengan akhir baris. Sehingga bila fungsi ini akan diberlakukan untuk data cuaca harus ada fungsi tambahan dimana pointer akan menunjuk pada indeks array tertentu sesuai dengan kebutuhan operator, pada posisi mana akan operator akan menguji deret data text cuaca tersebut.

Dari ketiga fungsi deteksi kesalahan yang memiliki peluang pendeteksian kesalahan lebih cepat adalah *First-Come*, fungsi ini sesuai dengan jenis data cuaca dimana intentitas waktu kedatangan data setiap 10 menit sekali, sehingga begitu data text berikutnya datang untuk dideteksi dapat dilakukan lebih cepat. Tingkat akurasi data fungsi *First-Come* memiliki nilai yang baik, hal ini berdasarkan hasil pengujian terhadap data yang masuk. (lihat tabel 4)

Tabel 4. Perbandingan Waktu Proses dan Akurasi Data

Fungsi	Rata-Rata Waktu Proses (detik)	Rata-Rata Akurasi Data (%)
wait-for-all	435	93,2
first-come	299	91,5
every-time	412	87,6

Pada tabel 4 dapat dilihat bahwa akurasi data untuk fungsi *wait-for-all* dan *first-come* memiliki nilai persentasi yang hampir sama. Nilai ini diperoleh dari 25 kali pengujian dengan jumlah data 200 data text. Proses pengujian akurasi data dilakukan dengan cara membandingkan data yang sudah tersimpan dalam sintaks SQL dengan data yang tersimpan di data *logger*. Nilai akurasi data ini tidak memperhitungkan gangguan yang terjadi pada proses pengiriman data melalui jaringan nirkawat.

V. KESIMPULAN

Proses tranformasi data text cuaca ke dalam sintaks SQL membutuhkan suatu proses deteksi kesalahan data. Proses pendeteksian kesalahan dengan menggunakan notasi EPC dan PetriNet menghasilkan suatu nilai pengukuran kecepatan waktu proses deteksi kesalahan dan akurasi data. Hal ini dibuktikan dengan pengujian uji terhadap data text yang tersimpan di basisdata cuaca dalam sintaks SQL. Data dalam sintkas SQL ini dapat digunakan untuk menampilkan informasi cuaca dan untuk sistem peringatan dini banjir.

DAFTAR PUSTAKA

- Fettke, P., & Loos, P, 2003, Classification of Reference Models - a Methodology and its Application. Journal of Information Systems and e-Business Management, p:202–211, Elsevier.
- Dumas, van der Aalst, and Hofstede , 2005, Process-Aware Information Systems. John Wiley & Sons, Inc.
- Mendling, J., 2008, Detection and prediction of errors in EPCs of the SAP reference model, Journal of Data & Knowledge Engineering 64, p : 312–329, Elsevier.
- Rosemann, M., 2007, A configurable reference modelling language, Journal of Information Systems 32, p:1–23, Elsevier.
- Recker, Jan, 2006, Configurable Reference Modeling Languages, Journal of Information Systems 24, p:45–53, Elsevier.
- Van der Aalst, Wil, Dreiling, Alexander, Gottschalk, Florian, Rosemann, Michael, & Jansen-Vullers, Monique, 2006, Configurable Process Models as a Basis for Reference Modelin, Business Process Management Workshops (LNCS 3812), pp. 512-518.
- Eshuis, Rik, and Wieringa, Roel, 2003, Comparing Petri Net and Activity Diagram Variants for Workflow Modelling A Quest for Reactive Petri Nets, Petri Net Technology for Communication Based Systems, Springer Link, Volume 2472, 2003, pp 321-351