

Xpath Dokumen Xml untuk Perolehan Data pada Basisdata Perundangan Pemerintah Kota Semarang

Setyawan Wibisono

Program Studi Teknik Informatika, Universitas Stikubank

email: setyasonny@gmail.com

Abstrak

Saat ini sebagian dokumen perundangan Pemerintah Kota Semarang masih banyak dalam bentuk hardcopy, belum dapat secara mudah diakses masyarakat melalui teknologi internet. Produk perundangan Pemerintah Kota Semarang yang dimaksud di sini adalah Peraturan Daerah Kota Semarang, Peraturan Walikota dan Surat Edaran Walikota

Dalam penelitian ini ada beberapa langkah yang dilakukan, yaitu: merancang XPath dengan memperhatikan node yang akan diambil. Xpath ini akan dapat diimplementasikan dalam XQuery untuk memperoleh data pada dokumen dalam format XML dari suatu koleksi dokumen perundangan Pemerintah Kota Semarang.

Program aplikasi untuk perolehan data pada basisdata perundangan Pemerintah Kota Semarang adalah memanfaatkan akses node pada sebuah XPath dokumen XML dalam membuat basisdata XML serta menggunakan XPath untuk memperoleh informasi sesuai query yang dimasukkan. Query yang dimasukkan adalah suatu kalimat dalam bahasa Indonesia. Sistem akan membuat sebuah jalur (path) yang disebut Xpath untuk mengakses node yang dituju sesuai dengan kalimat yang diinputkan.

Cara memperoleh data menggunakan cara menginputkan suatu kalimat dalam bahasa Indonesia, dengan tata bahasa yang telah ditentukan. Pada aplikasi ini juga diberikan contoh cara memberikan kalimat input, maupun pernyataan perintah untuk memperoleh data. Output dari aplikasi ini dalam bentuk tabel kepada user, jika input kalimat yang diberikan sudah benar dan data tersedia sesuai permintaan. Data yang ditampilkan adalah node dari hasil pengambilan data berdasarkan XPath. Jika format kalimat tidak sesuai dengan aturan maka yang dimunculkan oleh aplikasi ini adalah sebuah peringatan kesalahan.

Kata kunci : node, perundangan, XML, XQuery, XPath

PENDAHULUAN

Sampai saat ini informasi tentang perundangan masih relatif sulit untuk ditemukan, padahal perundangan merupakan suatu dasar bagi pemerintah dalam menjalankan roda pemerintahan serta masyarakat dalam menjalankan usaha dan urusan sehari-hari. Memang secara formal informasi tentang perundangan haruslah dapat diakses oleh masyarakat umum, tetapi dengan kenyataan yang sekarang terjadi, maka hal tersebut masihlah jauh dari kenyataan. Sebagai

gambaran bahwa arsip-arsip perundangan pada pemerintahan di daerah masih relatif sulit diperoleh, karena arsip-arsip dapat berada di unit kerja yang berlainan, bukan pada satu unit yang mengurus arsip perundangan. Walaupun pemerintah di daerah sudah mulai terbuka dengan mempublikasi arsip perundangan di daerah dengan membentuk jaringan dokumen dan informasi pada biro hukum di daerah. Maka diperlukan suatu upaya untuk menyimpan arsip-arsip perundangan secara

lebih tertib dan terdokumentasi secara elektronik sehingga mudah dalam melakukan publikasi.

Saat ini sebagian dokumen perundangan Pemerintah Kota Semarang masih banyak dalam bentuk hardcopy, belum dapat secara mudah diakses masyarakat melalui teknologi internet. Produk perundangan Pemerintah Kota Semarang yang dimaksud di sini adalah Peraturan Daerah Kota Semarang, Peraturan Walikota dan Surat Edaran Walikota

Pasal 18 ayat (6) UUD 1945 menyatakan bahwa pemerintah daerah berhak menetapkan peraturan daerah dan peraturan-peraturan lain untuk melaksanakan otonomi daerah dan tugas pembantuan. Dalam kaitan ini maka sistem hukum nasional memberikan kewenangan atributif kepada daerah untuk menetapkan Perda dan peraturan daerah lainnya, dan Perda diharapkan dapat mendukung secara sinergis program-program Pemerintah di daerah. Perda sebagaimana PUU lainnya memiliki fungsi untuk mewujudkan kepastian hukum (*rechtszekerheid, legal certainty*) [5].

Beberapa permasalahan yang dapat dirumuskan dalam penelitian ini adalah :

1. Bagaimana merancang dokumen dalam format XML dari suatu koleksi dokumen perundangan Pemerintah Kota Semarang.
2. Bagaimana merancang XPath yang dapat diimplementasikan dalam XQuery, sehingga dapat digunakan untuk perolehan informasi dokumen perundangan Pemerintah Kota Semarang.

Extensible Markup Language (XML) adalah format berbasis teks sederhana untuk mewakili informasi terstruktur: dokumen, data, konfigurasi, buku, transaksi, faktur, dan banyak lagi. Format ini berasal dari format standar yang lebih ada sebelumnya yang disebut SGML (ISO 8879), agar lebih sesuai digunakan untuk Web. XML adalah salah satu format yang paling banyak digunakan untuk berbagi informasi yang terstruktur hari ini: antar program, antar manusia, antara komputer

dan manusia, baik lokal maupun melalui jaringan.

XML Schema adalah bahasa untuk mengekspresikan batasan tentang dokumen XML. Ada beberapa bahasa skema yang berbeda digunakan secara luas, tapi yang utama adalah Document Type Definition (DTD), Relax-NG, Schematron dan W3C XSD (XML Schema Definition) [4].

XPath menggunakan jalur ekspresi untuk memilih node atau node-set dalam dokumen XML. Node dipilih dengan mengikuti jalur atau langkah. XPath adalah sintaks untuk mendefinisikan bagian dari dokumen XML. Ekspresi XPath paling berguna adalah jalur lokasi. Sebuah jalur lokasi menggunakan setidaknya satu langkah lokasi untuk mengidentifikasi satu set node dalam dokumen. Set ini mungkin kosong, berisi node tunggal, atau mengandung beberapa node. Node ini bisa menjadi elemen, atribut, namespace, teks, komentar, pemrosesan instruksi, node root, atau kombinasi dari mereka. Path lokasi yang paling sederhana adalah salah satu yang memilih simpul akar dokumen. Jalan ini hanya garis miring /. Akan terlihat bahwa banyak sintaks XPath memang sengaja dipilih untuk menjadi mirip dengan sintaks yang digunakan oleh shell Unix. Simbol / adalah akar dari file system Unix dan / adalah node root dari suatu dokumen XML. Misalnya, template XSLT menggunakan pola XPath / untuk mencocokkan pohon masukan seluruh dokumen dan membungkusnya dalam sebuah elemen html [2].

Dengan XQuery, baik pemetaan berbasis tabel atau object-relational mapping dapat digunakan. Jika pemetaan berbasis tabel yang digunakan, setiap tabel diperlakukan sebagai dokumen terpisah dan bergabung antara tabel (dokumen) yang ditentukan dalam permintaan sendiri, seperti di SQL. Jika object-relational mapping digunakan, hirarki tabel diperlakukan sebagai satu dokumen dan bergabung ditentukan dalam pemetaan. Tampaknya mungkin bahwa berbasis tabel pemetaan akan digunakan dalam implementasi yang paling

atas database relasional, karena ini tampaknya mudah untuk diterapkan dan lebih akrab bagi pengguna SQL [7].

CARA PENELITIAN

Dalam membangun aplikasi ini, langkah yang dilakukan adalah melakukan studi pustaka studi pustaka yang berhubungan dengan analisis data, pemodelan sistem dan perancangan sistem aplikasi, dengan pengumpulan data dari bahan – bahan referensi, arsip, dan dokumen yang berhubungan dengan permasalahan dalam penelitian ini. Langkah berikutnya adalah perancangan data dengan melakukan pengamatan, pencatatan, perancangan dan pembuatan basisdata dokumen perundangan Pemerintah Kota Semarang dengan format data XML.

Penelitian ini menggunakan model *prototyping*. Di dalam model ini sistem dirancang dan dibangun secara bertahap dan untuk setiap tahap pengembangan dilakukan percobaan-percobaan untuk melihat apakah sistem sudah bekerja sesuai dengan yang diinginkan.

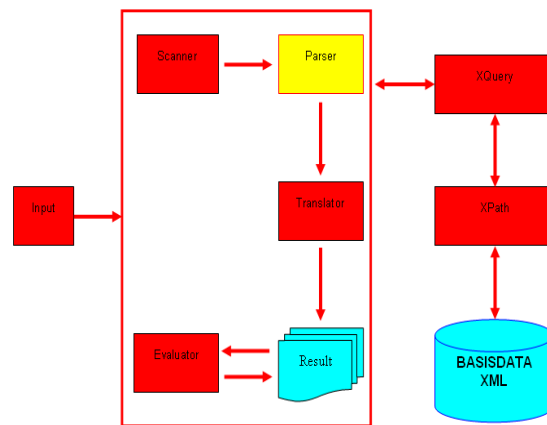
HASIL DAN PEMBAHASAN

1. Deskripsi Sistem

Program aplikasi untuk perolehan data pada basisdata perundangan Pemerintah Kota Semarang adalah memanfaatkan akses node pada sebuah XPath dokumen XML dalam membuat basisdata XML serta menggunakan XPath untuk memperoleh informasi sesuai query yang dimasukkan. Query yang dimasukkan adalah suatu kalimat dalam bahasa Indonesia. Sistem akan membuat sebuah jalur (path) yang disebut Xpath untuk mengakses node yang dituju sesuai dengan kalimat yang diinputkan. Aplikasi ini dapat digunakan oleh praktisi hukum, pimpinan bagian hukum, staf bagian hukum dan staf administrasi lainnya pada Pemerintah Kota Semarang. akademik untuk membantu pekerjaan pencarian data dalam format XML yang berkaitan dengan data perundangan Pemerintah Kota Semarang. Aplikasi ini dapat dipergunakan sebagai alat bantu dalam

pencarian data dengan domain perundangan pada Pemerintah Kota Semarang.

Dalam proses pencarian data pada sebuah basisdata, metoda yang lazim digunakan adalah menggunakan bahasa SQL untuk memperoleh data pada suatu tabel yang terdapat dalam suatu basisdata. Cara seperti ini adalah suatu cara yang banyak dilakukan oleh administrator dan operator suatu sistem yang melibatkan basisdata pada suatu sistem. Tetapi perolehan informasi dengan cara seperti ini hanya dipahami oleh orang yang mempunyai pengetahuan dan keterampilan dalam mengakses basisdata menggunakan perintah SQL.



Gambar 1. Akses Node Dokumen XML Menggunakan XPath

Pada aplikasi ini cara memperoleh data menggunakan cara menginputkan suatu kalimat dalam bahasa Indonesia, dengan tata bahasa yang telah ditentukan. Pada aplikasi ini juga diberikan contoh cara memberikan kalimat input, maupun pernyataan perintah untuk memperoleh data.

Prinsip kerja dari aplikasi ini adalah menerima input berupa kalimat. Kalimat dengan format yang benar maupun format yang keliru akan tetap menghasilkan tampilan. Kalimat input akan dipisahkan menjadi kata – kata. Pemisahan hanya berdasarkan urutan kata yang diinputkan. Dari hasil pemisahan kalimat akan muncul kumpulan kata. Kata – kata yang muncul akan dibandingkan dengan kata – kata yang sudah terdaftar. Perbandingan kata dengan daftar kata akan memunculkan

dua kelompok kata yaitu kelompok kata yang bermakna (token) dan kelompok kata yang tidak mempunyai makna. Kata yang tidak bermakna akan disingkirkan dan tidak dilakukan proses terhadapnya. Sedangkan kata yang bermakna akan dibandingkan dengan daftar kata yang termasuk atribut, pelengkap, operator dan kondisi. Dari kumpulan kata – kata yang penting, akan dikonstruksikan menjadi XPath sesuai dengan aturan produksi pembentuk kalimat. Jika sudah sesuai dengan aturan produksi, maka akan diterjemahkan dalam XQuery untuk mengakses basisdata XML, sehingga menghasilkan keluaran field yang sesuai dengan node yang dituju. Keluaran yang dihasilkan adalah sebuah tabel yang muncul pada sebuah form. Tabel tersebut berisi tampilan data perundangan yang sesuai dengan apa yang ditanyakan atau diperintahkan oleh user.

2. Struktur Data

Dalam proses koleksi data maka langkah awal yang dilakukan adalah penyimpanan data dengan format XML. Data perundangan dibuat dalam XML schema hal ini dimaksudkan untuk memastikan bahwa elemen – elemen dan atribut – atribut yang dimasukkan ke dalam dokumen sudah memenuhi aturan yang diterapkan dalam skema. XML Schema TblPerundangan untuk data yang disimpan adalah sebagai berikut :

1. jenisperundangan
2. nomor
3. tahun
4. isi :
 - a. namaperundangan
 - b. pejabat pembuat
 - c. penyetuju1
 - d. penyetuju2
 - e. disahkan
 - f. diundangkan

Penggambaran data dengan format XML pada XML Schema akan terlihat seperti gambar 2. Field yang dimiliki dokumen

perundangan akan diperlihatkan sebagai field utama dengan tipe data string, sedangkan field yang dimiliki oleh isi diperlihatkan sebagai sebuah susunan yang bersarang dengan tipe data string.

Gambar 2. XML Schema

Penggambaran data dengan format XML pada XML Grid akan terlihat seperti gambar 3. Field yang dimiliki dokumen akan diperlihatkan sebagai elemen utama, sedangkan field yang dimiliki oleh isi diperlihatkan sebagai elemen dengan susunan yang bersarang.

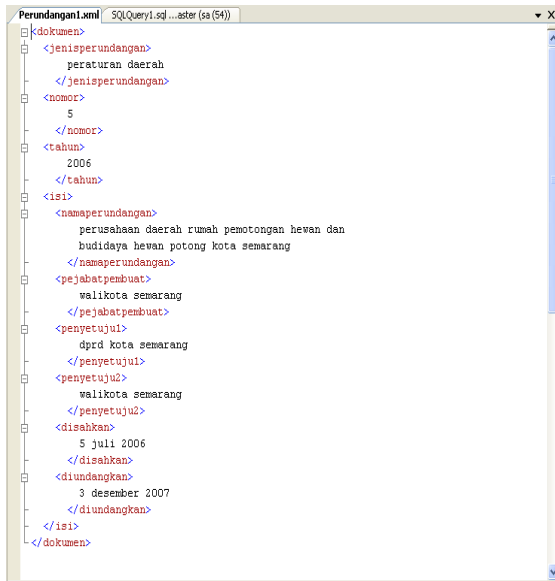
Field	Value												
jenisperundangan	peraturan daerah												
nomor	5												
tahun	2008												
isi	<table border="1"> <tbody> <tr> <td>namaperundangan</td> <td>perubahan daerah nama pendongkaran hewan dan budidaya hewan potong lida semarang</td> </tr> <tr> <td>pejabat pembuat</td> <td>walikota semarang</td> </tr> <tr> <td>penyetuju1</td> <td>dprd kota semarang</td> </tr> <tr> <td>penyetuju2</td> <td>walikota semarang</td> </tr> <tr> <td>disahkan</td> <td>5 Juli 2008</td> </tr> <tr> <td>diundangkan</td> <td>3 Desember 2007</td> </tr> </tbody> </table>	namaperundangan	perubahan daerah nama pendongkaran hewan dan budidaya hewan potong lida semarang	pejabat pembuat	walikota semarang	penyetuju1	dprd kota semarang	penyetuju2	walikota semarang	disahkan	5 Juli 2008	diundangkan	3 Desember 2007
namaperundangan	perubahan daerah nama pendongkaran hewan dan budidaya hewan potong lida semarang												
pejabat pembuat	walikota semarang												
penyetuju1	dprd kota semarang												
penyetuju2	walikota semarang												
disahkan	5 Juli 2008												
diundangkan	3 Desember 2007												

Gambar 3. XML Grid

3. Implementasi Basisdata XML

Ketika basisdata produk perundangan Pemerintah Kota Semarang akan digunakan, maka harus dibuat terlebih dahulu. Sehingga langkah awal dalam implementasi sistem ini adalah merancang dan mengimplementasikan basisdata perundangan Pemerintah Kota Semarang sebagai basisdata XML terlebih

dahulu. Data produk perundangan Pemerintah Kota Semarang ditampilkan dalam struktur data seperti ditampilkan dalam gambar 4.



Gambar 4. Struktur data XML Perundangan Pemerintah Kota Semarang

Data perundangan Pemerintah Kota Semarang dalam format XML diimplementasikan menggunakan SQL Server 2008. Gambar 5 adalah tampilan data perundangan Pemerintah Kota Semarang dalam format XML.



Gambar 5. Tampilan data perundangan dalam format XML

4. Koneksi Database

Pada subprogram ini ditentukan koneksi aplikasi dengan database dan tabel yang digunakan dalam aplikasi ini. Database yang digunakan adalah "DBWALIKOTA" dan tabel yang digunakan adalah "TblPerundangan", dengan username "sa" dan password "aliya", seperti terlihat pada gambar 6

```
public class DisplayQueryWlp extends JFrame
{
    // JDBC driver and database URL
    static final String JDBC_DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver";
    static final String DATABASE_URL = "jdbc:odbc:TblPerundangan";
    static final String DATABASE_URL = "jdbc:odbc:DBWALIKOTA";
    static final String USERNAME = "sa";
    static final String PASSWORD = "aliya";

    // default query selects all rows from authors table
    static final String DEFAULT_QUERY = "select Perundangan.value('dokumen[1]/jenisperundangan[1]";

    private ResultSetTableModel tableModel;
    private JTextArea queryArea;

    // create ResultSetTableModel and GUI
    public DisplayQueryWlp()
    {
        super ( "AKSES NODE XPATH " );

        // create ResultSetTableModel and display database table
        try
        {
            // create TableModel for results of query SELECT * FROM authors

```

Gambar 6. Koneksi database

5. Implementasi Program

Ketika aplikasi dijalankan akan ditampilkan sebuah default query bahasa Indonesia yang sesuai dengan format kalimat. Jika format query sudah benar, tetapi data yang dikehendaki tidak tersedia, maka akan ditampilkan tabel kosong. Begitu juga jika terdapat kesalahan dalam penulisan format query, maka akan ditampilkan pesan kesalahan, seperti terlihat pada gambar 7.

```
catch ( SQLException sqlException )
{
    JOptionPane.showMessageDialog( null,

    //sqlException.getErrorCode(),
    sqlException.getMessage(),
    "Database Error Query Tidak berhasil ",
    JOptionPane.ERROR_MESSAGE );

    // try to recover from invalid user query
    // by executing default query
    try
    {
        tableModel.setQuery( DEFAULT_QUERY );
        queryArea.setText( DEFAULT_QUERY );
    } // end try
    catch ( SQLException sqlException2 )
    {
        JOptionPane.showMessageDialog( null,
        sqlException2.getMessage(), "Database Error ",
        JOptionPane.ERROR_MESSAGE );
    }
}
```

Gambar 7. Pesan kesalahan

Scanner

Dalam implementasi sistem ini, scanner yang digunakan adalah utilitas scanner yang ada dalam library Java. Untuk menggunakannya cukup dengan memanggilnya, dengan proses pemanggilan diimplementasikan seperti terlihat pada gambar 8.

```
package nlp.parser;
import java.util.Scanner;
import java.util.regex.Pattern;
```

Gambar 8. Penggunaan Utility Scanner

Parser

Analisis sintaks dilakukan berdasarkan aturan produksi yang telah ditentukan dan akan menghasilkan pohon sintaks (pohon parser). Seperti terlihat pada gambar 9 proses perubahan pohon sintaks hasil parsing ke dalam tipe – tipe query diimplementasikan dalam bagian program ini.

```
public class Parsing
{
    public String Pengurai(String str)
    {
        String names[] = { "jenis", "jenisperundangan", "nama", "namaaperundangan", "pejabat", "pejabatpen
SortedSet<String> treex = new TreeSet<String>( Arrays.asList( names ) );

List list
List listkon
List listekor
String tdan
String koma
boolean frasekondisi
boolean adeakor
String tok, tokenlist;
String bantu
String bantuan
String kkekunci
StringBuffer sbf
StringBuffer sfbantu

String regex = "[di]ana[sa]ja[saj]a[ta]mpilkan[ta]mpil[cari]daftar[te]ngandung[di]pada[dari]yang[o]
```

Gambar 9. Deklarasi Parser

Translator

Pohon sintaks yang dihasilkan dari proses parsing kemudian diterima translator. Translator berfungsi menghasilkan kode antara, sebelum memasuki proses evaluasi untuk menentukan jawaban akhir query. Komponen evaluator berfungsi mengevaluasi daftar goal dalam proses penentuan jawaban query, sehingga evaluasi banyak terlibat dengan entri – entri basisdata, seperti terlihat pada gambar 10.

```
public class Evaluator
{
    private List fraseatribut;
    private List frasekondisi;
    private StringBuffer strbf=new StringBuffer("select distinct ");
    private Object bantu, bantuan;

    // Variabel bersama querysatu dan querydua
    String idan
    String istau
    String ikama
    String sambung[]
    StringBuffer strbfr
    String ekorkom
    String ekorkomp
    String utaaa[]
    String isa[]

SortedSet<String> treestaaa = new TreeSet<String>( Arrays.asList( utaaa ) );
SortedSet<String> treestisi = new TreeSet<String>( Arrays.asList( isa ) );
SortedSet<String> treestabung = new TreeSet<String>( Arrays.asList( sambung ) );

// Variabel khusus querydua
String kkrang
String fibah
String suka
String sukat
Boolean statkom
String tolak[]
String danatau[]
SortedSet<String> treestolak
SortedSet<String> treedanatau
int posisi;
```

Gambar 10. Translator

Evaluator

Baris pertama akan mengambil satu simpul terdepan dari linked list token untuk dimasukkan ke dalam variabel bantu. Baris kedua akan menguji apakah himpunan treeutama mengandung string yang dimuat dalam variabel bantu. Jika ya, maka kemudian dilakukan penterjemahan ke dalam fragmen sintaks XQuery.

Untuk XML dengan data bersarang, proses perubahan pohon sintaks hasil parsing ke dalam tipe – tipe query diimplementasikan dalam bagian program seperti terlihat pada gambar 11. Setelah dilakukan penguraian berdasarkan frase atribut dan frase kondisi. Selanjutnya dilakukan penguraian berdasarkan masing – masing frase.

```
.....
// Mulai querysatu
public String querysatu(List fratribut, List frakondisi)
{
    while (fratribut.isEmpty() != true)
    {
        bantu=fratribut.removeFromFront();
        if (treeutama.contains(bantu)) //atribut kode, matakuliah, progdi dll
        {
            strbf.append(" Perundangan.value('/dokumen[1]/")
                .append(bantu)
                .append("[1]', 'varchar(200)') as ")
                .append(bantu);
            continue;
        }

        if (treeisi.contains(bantu)) // atribut deskripsi
        {
            strbf.append("a.value('");
            strbf.append(bantu)
                .append("[1]', 'varchar(200)') as ")
                .append(bantu);
            ekorkomp="CROSS APPLY Perundangan.nodes('/dokumen/isi') as result(a)";
            continue;
        } // atribut deskripsi
```

Gambar 11. Evaluator

Contoh Pertanyaan

Program DemoFrame ini digunakan untuk memperlihatkan pola pertanyaan sekaligus beserta contoh kalimat masukan yang dapat memberikan jawaban dengan benar, seperti terlihat pada gambar 12.


```

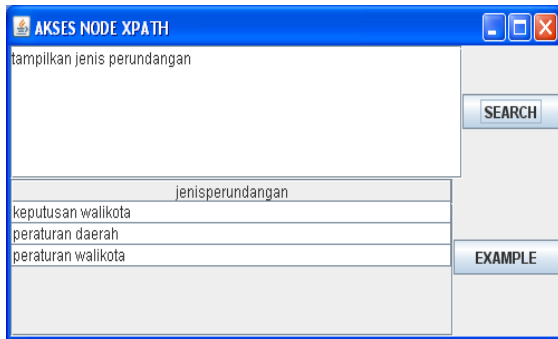
public DemoFrame()
{
    queryArea = new JTextArea
    {
        "POLA PERNYATAAN INPUT BESERTA CONTOH \n \n" +
        "===== \n \n" +
        "POLA 1 : QUERY - ATRIBUT \n" +
        "    Tampilkan jenis perundangan \n \n" +
        "POLA 2 : QUERY - ATRIBUT - ATRIBUT \n" +
        "    Tampilkan jenis perundangan tahun \n \n" +
        "POLA 3 : QUERY - ATRIBUT - ATRIBUT - OPERATOR \n" +
        "    Tampilkan jenis perundangan nomor dan tahun dengan tahun 2006 \n \n" +
        "POLA 4 : QUERY - ATRIBUT - ATRIBUT - OPERATOR - ATRIBUT \n" +
        "    Tampilkan jenis perundangan nomor dengan pejabat pembuat walikota dan tahun dengan
        "POLA 5 : QUERY - ATRIBUT - OPERATOR - DATA \n" +
        "    Tampilkan jenis perundangan nomor dan tahun dengan tahun lebih dari 2006 \n \n" +
        "POLA 6 : QUERY - ATRIBUT - BUKAN - DATA \n" +
        "    Tampilkan jenis perundangan dengan tahun bukan 2006 \n \n" +
        "POLA 7 : QUERY - ATRIBUT - ATRIBUT - BUKAN - DATA \n" +
        "    Tampilkan jenis perundangan nomor dan tahun dengan tahun bukan 2006 \n \n" +
        "===== \n \n" +
        "20, 300);
        queryArea.setWrapStyleWord(true);
        queryArea.setLineWrap(true);
        queryArea.setEditable(false);
    }

    public void createAndShowGUI()
    {
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("BERBERAPA POLA DAN CONTOH PERNYATAAN INPUT");
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        frame.getContentPane().add(queryArea);
    }
}
    
```

Gambar 12. Contoh Pertanyaan

6. Tampilan Aplikasi

a) Tampilan Tipe q_a (query – atribut)



Gambar 13. Hasil Pertanyaan Satu Atribut

Pertanyaan atau masukan : **“Tampilkan jenis perundangan”**. Akan ditampilkan hasil sesuai pertanyaan, yaitu semua jenis perundangan yang terdapat dalam tabel XML seperti terlihat pada gambar 13.

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut [1,3,6] :

1. Scanner : [“jenis perundangan”]

Kalimat pertanyaan atau masukan dipecah menjadi kata-kata, yaitu “tampilkan” dan “jenis perundangan”. Kemudian daftar kata tersebut akan dicocokkan dengan daftar token. Kata yang termasuk token akan diambil, yaitu “jenis perundangan” sedangkan kata yang tidak termasuk token, yaitu “tampilkan” akan dibuang.

2. Parser : query(fraseatribut(“jenis perundanga ”,kosong))

Hasil pemilahan token akan dimasukkan ke dalam parser untuk disesuaikan dengan analisis sintaks. Tipe sintaks ini meminta untuk sebuah atribut tanpa adanya kondisi yang mengikutinya.

3. Translator : q_a(“jenis perundangan”)

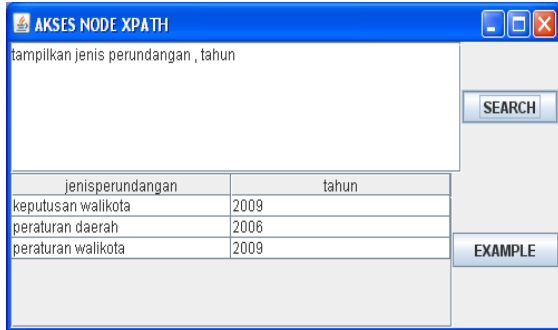
Hasil analisis sintaks akan dilakukan pencocokan dengan pohon sintaks sehingga akan didapatkan query yang tepat. Hasil pencarian nama atribut yang diperoleh adalah ”jenis perundangan”

Hasil parser diolah oleh translator dan terjadi proses pemadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian nama atribut dari data yang diketahui, dan diperoleh atribut “jenis perundangan”.

4. Evaluator

Dari tabel yang ada, maka dicari atribut dalam tabel yang sesuai dengan hasil translator. Dalam kasus ini atribut terdiri dari dua buah kata yang dipisahkan oleh spasi, sehingga kemungkinan akan terjadi ketidaksesuaian jika langsung mencocokkan kata pertama dengan atribut dalam tabel. Maka dalam tahap ini kata yang menjadi atribut harus dicari kata pengikutnya, jika tidak ditemukan kata pengikutnya, maka kata itu akan langsung dicocokkan dengan atribut dalam tabel. Untuk kata yang memiliki kata pengikut, maka harus dicari kata pengikutnya. Setelah ditemukan kata pengikutnya, maka 2 kata atau lebih tersebut akan dicocokkan dengan atribut dalam tabel, sehingga ditemukan field dalam tabel yang cocok, sehingga field tersebut dapat ditampilkan.

- b) Tampilan Tipe q_a_a (query – atribut – atribut)



Gambar 14. Hasil Pertanyaan Lebih dari Satu Atribut

Pertanyaan atau masukan : **”Tampilkan Jenis Perundangan, Tahun”**

Tipe pertanyaan ini meminta tampilan data yang terdiri dari dua atribut atau lebih yang dipisahkan oleh tanda koma. Pada contoh query ini meminta untuk menampilkan data jenis perundangan dan tahun yang terdapat dalam tabel seperti terlihat pada gambar 14.

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut [1,3,6] :

1. Scanner : [“jenis perundangan”, ””, ”tahun”]

Kalimat pertanyaan atau masukan dipecah menjadi kata-kata, yaitu “tampilkan”, “jenis perundangan” dan “tahun”. Kata yang tidak bermakna yaitu “tampilkan” akan dibuang. Tanda baca “,” (koma) akan dipahami oleh program sebagai sebuah kata sambung. Kata sambung yang dipahami oleh program adalah “dan” dan “koma”. Token yang dipahami oleh program adalah “jenis perundangan” dan “tahun”.

2. Parser : query(fraseatribut(“jenis perundangan”, ”tahun”, kosong))

Pemecahan kata akan diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta atribut “jenis perundangan”, “tahun” tanpa frase kondisi.

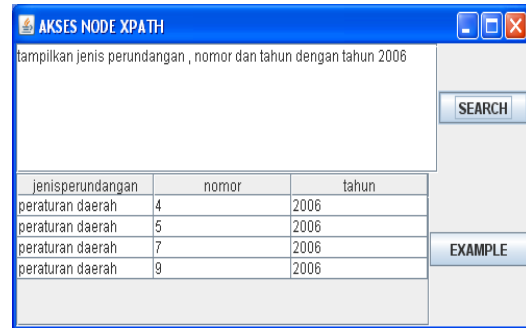
3. Translator : q_aa(“jenis perundangan”, “tahun”)

Hasil parser diolah oleh translator dan terjadi proses pepadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian nama atribut dari data yang diketahui, dan diperoleh atribut “jenis perundangan”, “tahun”.

4. Evaluator

Dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya/perintah menjadi suatu pernyataan XQuery. Pernyataan XQuery tersebut kemudian dikoneksikan dengan atribut dalam tabel di database DBWALIKOTA, sehingga akan menampilkan data sesuai dengan masukan.

- c) Tampilan Tipe q_a_opr (query – atribut – operator)



Gambar 15. Hasil Pertanyaan dengan Sebuah Kondisi yang Diketahui

Pertanyaan atau masukan : **”Tampilkan jenis perundangan, nomor dan tahun dengan tahun 2006”**

Tipe query ini adalah query untuk meminta data yang berupa beberapa atribut sekaligus dari tabel. Atribut yang hendak ditampilkan dapat terdiri dari 2 atribut atau lebih dengan sebuah kondisi yang diketahui. Pada contoh query ini meminta aplikasi untuk menampilkan data ”jenis perundangan”, ”nomor”, ”tahun” dengan tahun yang mengandung kondisi ”2006” seperti terlihat pada gambar 15.

Tahapan pemrosesan untuk masukan tersebut sebagai berikut [1,3,6] :

1. Scanner : [“jenis perundangan”, “nomor”, ”tahun”, ”tahun”, ”2006”. Kalimat pertanyaan atau masukan dipecah menjadi kata-kata, yaitu “tampilkan”, “jenis perundangan”, ”nomor”, “tahun”, ”dengan”, ”tahun”, ”2006”. Kemudian kata “tampilkan” dan ”dengan” dibuang karena termasuk dalam kata – kata yang dapat diabaikan. Diperoleh kata ”jenis perundangan”, ”nomor”, “tahun” sebagai kata yang bermakna (token), serta aribut “tahun” berikutnya sebagai kondisi, dan “2006” sebagai data.

2. Parser :

query(fraseatribut(“jenis perundangan”, kosong), (“nomor”, kosong), fkondisi(“tahun”, data(“2006”, kosong)))

Hasil scanner diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta atribut “tahun” dengan frase kondisi “tahun” dengan data “2006”.

3. Translator : q_a_opr(“jenis perundangan”, “nomor”, “tahun”, “2006”)

Hasil parser diolah oleh translator dan terjadi proses pepadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian atribut “tahun” dari data yang diketahui “2006”

4. Evaluator

Dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya/perintah menjadi suatu pernyataan XQuery. Pernyataan XQuery tersebut kemudian dikoneksikan dengan atribut dalam tabel di database DBWALIKOTA, sehingga akan menampilkan data sesuai dengan masukan.

d) Tampilan Tipe q_a_bukan (query – atribut – bukan – data)

jenisperundangan	nomor	tahun
keputusan walikota	810/357	2009
peraturan walikota	13	2009
peraturan walikota	30	2009

Gambar 16. Hasil Pertanyaan dengan Sebuah Kondisi yang Diketahui

Pertanyaan atau masukan : ”**Tampilkan jenis perundangan, nomor dan tahun dengan tahun bukan 2006**”

Tipe query ini adalah query untuk meminta data yang berupa beberapa atribut sekaligus dari tabel. Atribut yang hendak ditampilkan dapat terdiri dari dua atribut atau lebih dengan sebuah kondisi yang diketahui dan operator “bukan”. Pada contoh query ini meminta aplikasi untuk menampilkan data ”jenis perundangan”, ”nomor”, ”tahun” dengan tahun yang mengandung kondisi ”2006” dan operator “bukan” seperti terlihat pada gambar 16..

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut [1,3,6]:

1. Scanner : [“jenis perundangan”, “nomor”, “tahun”, “tahun”, “bukan”, “2006”]

Masukan query diubah menjadi list kata – kata, yaitu “tampilkan”, “jenis perundangan”, “nomor”, “dan”, “tahun”, “dengan”, “tahun”, “bukan”, “2006”.

Kemudian kata “tampilkan”, “dengan”, dibuang karena termasuk dalam kata – kata yang bukan token. Diperoleh token yaitu “jenis perundangan”, “nomor”, “dan”, “tahun”, sebagai kata yang bermakna (token), serta kondisi “tahun”, operator “bukan” dan data “2006”.

2. Parser : fraseatribut(atribut, kosong), fkondisi(bukan, atribut, kosong))

query("jenis perundangan", "nomor", "tahun"), fkondisi("bukan", "2006")

Hasil scanner diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta atribut "jenis perundangan" dan "nomor" dengan frase kondisi "tahun", operator "bukan" dengan data "2006".

3. Translator : q_aa_opr("jenis perundangan", "nomor", "tahun", "bukan", "2006")

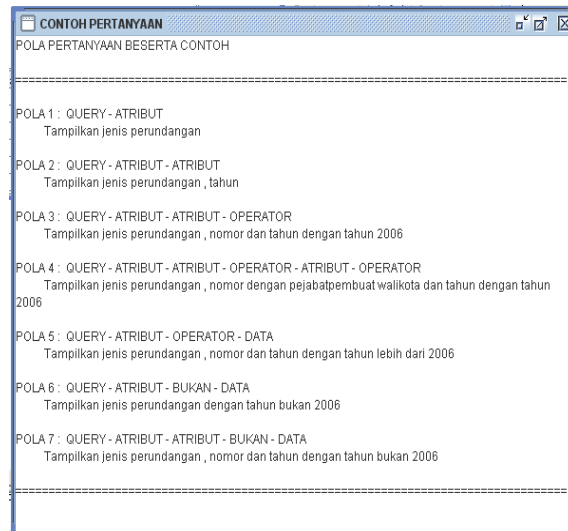
Hasil parser diolah oleh translator dan terjadi proses pemadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian atribut "jenis perundangan" dan "nomor" dengan kondisi "tahun", operator "bukan" dan data yang diketahui "2006".

4. Evaluator

Dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya / perintah menjadi suatu pernyataan XQuery, seperti terlihat pada gambar 16. Pernyataan XQuery tersebut kemudian dikoneksikan dengan tabel di database DBWALIKOTA.

- e) Tampilan contoh-contoh format kalimat input

Tampilan CONTOH PERTANYAAN ini digunakan untuk memperlihatkan pola pertanyaan sekaligus beserta contoh kalimat masukan yang dapat memberikan jawaban dengan benar, seperti terlihat pada gambar 17.



Gambar 17. Contoh Kalimat Masukan

KESIMPULAN

Berdasarkan hasil penelitian ini, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Aplikasi ini dapat digunakan sebagai mesin pencari untuk perolehan data produk perundangan Pemerintah Kota Semarang
2. Untuk memperoleh informasi, aplikasi ini menggunakan XPath untuk mendapatkan node atau data yang sesuai dengan permintaan. XPath tidak secara eksplisit terlihat dalam listing program maupun dalam tampilan, tetapi XPath diletakkan pada bagian program Translator. Pada Translator itulah sebenarnya peran XPath dalam mengakses node data.
3. Aplikasi ini memberikan contoh pertanyaan maupun perintah untuk mengakses node.
4. Aplikasi ini mampu memberikan keluaran berupa tabel yang berisi data. aplikasi ini juga mampu menampilkan XQuery hasil translasi dari kalimat masukan, dalam hal ini berupa pernyataan SQL dalam format XML.

DAFTAR PUSTAKA

- Andayani, S. (2002). Query Bahasa Indonesia untuk Basis Data Akademik, Tesis Magister Komputer, Program Pasca Sarjana Ilmu Komputer, UGM, Yogyakarta.
- Harold, E.R. dan Means, W.S. (2001). “*XML in a Nutshell A Desktop Quick Reference*”,
<http://oreilly.com/catalog/xmlnut/chapter/ch09.html>, diakses tanggal : 12 Maret 2012.
- Hartati, S. dan Zuliarso E., (2008). *Aplikasi Pengolah Bahasa Alami untuk Query Basisdata XML*, Dinamik, Vol XIII, No 2, Juli 2008, 168 – 175.
- Schema - W3C. (n.d.). *World Wide Web Consortium (W3C)*.
<http://www.w3.org/standards/xml/schema>. Diakses tanggal 12 Maret 2012
- Xmlquery (n.d.). rpbourret.com - XML and Databases. *rpbourret.com - XML consulting, writing, and research*.
<http://www.rpbourret.com/xml/XMLAndDatabases.htm/#xmlquery>. Diakses tanggal 12 Maret 2012.
- Murti, M.S. (2010). ”*Harmonisasi Peraturan Daerah Dengan Peraturan Perundang UndanganLainnya*,
<http://www.djpp.kemenkumham.go.id/htn-dan-puu/422-harmonisasi-peraturan-daerah-dengan-peraturan-perundang-undangan-lainnya.html> , diakses tanggal : 3 Maret 2012
- Wibisono, S. (2008). “*Aplikasi Pengolah Bahasa Alami untuk Query Basisdata Akademik dengan Format Data XML*”, Tesis Program S2 Ilmu Komputer, Universitas Gadjah Mada.
- Xmldatabase (n.d.). rpbourret.com - XML and Databases. *rpbourret.com - XML consulting, writing, and research*.
<http://www.rpbourret.com/xml/XMLAndDatabases.htm/#isxmlatabase>. Diakses tanggal 12 Maret 2012