



Implementasi Operasi Relasional *Extensible Markup Language (XML)* Pada Sistem Informasi

Mulyadi S.¹⁾, Armin Lawi²⁾, Andani Achmad³⁾

Jurusan Sistem Komputer STMIK Handayani, Makassar

¹mulyadicaak@gmail.com, ²armin@unhas.ac.id, ³andani60@yahoo.com

Abstrak

Dalam penelitian kami, kami fokus untuk mengimplementasikan Operasi Relasional *Extensible Markup Language (XML)* Pada Sistem Informasi. Penelitian ini bertujuan mengimplementasikan operasi aljabar relasional dalam pencarian data di dalam halaman website dan menganalisis penerapan struktur data pohon/tree dalam proses pencarian data dalam dokumen xml melalui path expression. Hasil penelitian ini menunjukkan bahwa, operasional aljabar relasional pada sistem data dapat diterapkan. Data yang telah disimpan kedalam format XML dapat diakses menggunakan operasi relasional, terutama untuk operasi pencarian yang merupakan inti dari sebuah aplikasi database. Sehingga sebagai pembandingan dibuatkan sebuah data pusat Mysql dengan struktur yang menyerupai relasi XML yang digunakan. Dari hasil percobaan yang dilakukan, rata-rata waktu query pencarian data menggunakan mysql adalah sebesar 0.0406 detik sedangkan waktu pencarian query menggunakan XML adalah sebesar 0.0393 detik. Ini berarti dengan menggunakan XML, dapat mengurangi waktu eksekusi hingga 1,03 kali dibandingkan dengan database client server seperti Mysql.

Kata Kunci : Aljabar Relasional, Query, Database, XML, Mysql

Abstract

In our research, we focus to implement Operation Relational Extensible Markup Language (XML) on Information Systems. This study aims to implement an operation of algebra relational in for finding data in inside the website and analyze the application of a structure of data tree / tree in the process for finding data in the document xml through path expression. This research result indicates that, operational algebra relational on a system data can be applied. The data that has been kept into format XML can be accessed use operation relational, especially for a search operation constituting the core of an application database. So that by contrast cut out for a central databank Mysql with structure that resembles relation XML used. From the experimts conducted, the average time query search data using Mysql is of 0.0406 seconds while search time query use XML are of 0.0393 seconds. This means using XML, reduces the time the execution until 1,03 times compared with a database client server as Mysql.

Keywords : Algebra Relational, Query, Database, XML, Mysql



1. Pendahuluan

Perguruan tinggi adalah salah satu tempat mahasiswa menuntut ilmu dan melanjutkan pendidikan ke tingkat yang lebih tinggi. Dengan berkembangnya suatu Perguruan tinggi, bertambahnya jumlah mahasiswa dan ilmu pengetahuan yang semakin berkembang mau tidak mau Perguruan tinggi harus meningkatkan pelayanan yang lebih efisien. Perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan atau karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka [1]. Sistem Informasi Perpustakaan adalah sistem yang dibuat untuk memudahkan petugas perpustakaan dalam mengelola suatu perpustakaan. Semua di proses secara komputerisasi yaitu digunakannya suatu software tertentu seperti software pengolah database. Penerapan teknologi informasi di perpustakaan dapat difungsikan dalam dua bentuk yaitu: (1) penerapan teknologi informasi digunakan sebagai sistem informasi manajemen perpustakaan, mulai dari pengadaan, inventarisasi, katalogisasi, sirkulasi bahan pustaka, pengelolaan anggota, statistik dan lain sebagainya, (2) penerapan teknologi informasi sebagai sarana untuk menyimpan, mendapatkan dan menyebarkan informasi ilmu pengetahuan dalam format digital [2].

Kenyataannya teknologi saat ini masih memiliki keterbatasan, yaitu : adanya kesulitan untuk dilakukan silang teknologi antar sistem operasi yang satu dengan sistem operasi yang lain, maupun antar satu bahasa pemrograman yang satu dengan bahasa pemrograman yang lain [3]. XML didesain sebagai solusi interoperabilitas antar aplikasi pada platform yang berbeda. Misalnya aplikasi A berjalan di atas platform Java, ingin berbagi informasi dengan aplikasi yang berjalan di atas platform .NET. aplikasi A akan membaca request dari aplikasi B, kemudian menyediakan informasi yang dibutuhkan oleh aplikasi B dalam format XML. Atau bisa jadi aplikasi A menyediakan informasi yang sudah dikemas dalam format XML, yang dapat dimanfaatkan oleh aplikasi B, C, D, dan seterusnya [4]. XML mampu menangani berbagai kompleksitas markup bertingkat-tingkat dan dapat ditukar/digabung dengan dokumen XML lain.

Aljabar relasional adalah sekumpulan operasi yang digunakan untuk melakukan proses manipulasi data dalam rangka untuk mendapatkan informasi yang diperlukan dari database. Aljabar lebih mudah untuk merancang, mengevaluasi, dan juga memperluas bahasa query bersama dengan query optimizer dan juga mesin eksekusi [5]. Bahasa query relasional adalah bahasa yang digunakan dalam basis data, sehingga dapat diterapkan untuk memperoleh informasi secara efisien.

Penelitian ini dimaksudkan untuk menganalisis implementasi aljabar relasional dalam pencarian data di dalam file XML yang didukung oleh teori-teori dan daya dukung atas kemajuan bahasa pemrograman sebagai media dalam pengekseskuan *Query* serta solusi efisiensi dan integrasi data dalam melakukan proses manipulasi informasi pada suatu dokumen.

2. Tinjauan Pustaka

2.1. XML (*Extensible Markup Language*)

XML (*eXtensible Markup Language*) merupakan suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa markup adalah mekanisme untuk mengenal suatu struktur didokumen. Tujuan dari XML secara umum untuk memungkinkan SGML untuk membantu, menerima dan memproses pada program web, dimana cara ini dapat dilakukan dengan HTML. XML mempunyai tujuan memungkinkan XML berfungsi sebagai suatu format pertukaran data (*data exchange*) antar dua entitas, XML juga dapat berperan dalam penyimpanan dokumen agar dapat digunakan oleh program lain, karena formatnya standar dan mampu mendeskripsikan data [6].

Adapun keuntungan menggunakan XML adalah [4] :

1. Extensibilitas : Dapat menentukan *tag-tag* sesuai dengan kebutuhan.
2. Dapat dijadikan alat pertukaran data dari sistem yang berbeda.
3. Memisahkan data dengan presentasi. Sebuah data hanya dapat berisi data saja tanpa ada informasi lain bagaimana cara menampilkannya. Karena dokumen XML menjelaskan datanya sendiri.
4. XML merupakan *file* teks sehingga bisa dilakukan pertukaran data melalui internet (HTTP).
5. Dengan satu *file* XML dapat dilakukan manipulasi tampilan sesuai dengan keinginan pengguna.
6. Proses *query* dari data XML lebih cepat karena bentuk struktur data yang berupa *tree* meskipun berbeda platform.
7. Dapat ditukar atau digabung dengan dokumen lain Dengan memisahkan isi data dengan presentasi, maka akan mempermudah penggabungan data dari berbagai macam sumber.
8. Penyederhanaan aplikasi. Untuk mengakses data menjadi lebih sederhana, *client* tidak perlu memerlukan lisensi atau hak akses terhadap data.

2.2. Aljabar Relasional

Terdapat 5 (lima) notasi aljabar relasional dasar yaitu, Seleksi (*selection*), Proyeksi (*projection*), Perkalian (*cartesian product*), Penggabungan Himpunan (*set union*) dan Selisih Himpunan (*set difference*) [7][8].

1. Seleksi (σ)

Seleksi adalah operasi untuk menyeleksi tupel tupel/ elemen-elemen yang memenuhi suatu predikat. Operator yang digunakan adalah:

- Operator perbandingan di gunakan untuk membandingkan data di suatu field dengan nilai. Operator ini terdiri atas : =, \leq , \geq , >, <.
- Beberapa predikat dapat di kombinasikan menjadi predikat mejemuk menggunakan penghubung 'and' (\wedge) and 'or' (\vee)

Operator proyeksi dinyatakan sebagai

$\sigma_{\text{selection predicate (Relation)}}$

dengan R menyatakan himpunan (relasi).

2. Proyeksi (π)

Proyeksi adalah memilih field (kolom) tertentu dari himpunan/subhimpunan dan membuang yang lain. Operasi proyeksi adalah operasi unary yang memroyeksikan kolom tertentu, dimana baris-baris duplikasi dihilangkan karena relasi merupakan suatu himpunan. Operator proyeksi dinyatakan sebagai

$\pi_{a_1, a_2, \dots, a_n}$ (Relation)

dengan a_1, a_2, \dots, a_n adalah list yang berisi satu atau lebih field yang ada pada himpunan (relasi) R.

3. Perkalian (\times)

Perkalian (\times) adalah operasi untuk menghasilkan dokumen hasil perkalian kartesian.

Operator Perkalian dinyatakan sebagai

$A \times B = \{(X, Y) \mid X \in A \text{ and } Y \in B\}$

dengan A akan dipasangkan dengan sebuah record di B dan hasil dari operasi tersebut akan memuat semua field yang ada di himpunan R dan S. Operator Cartesian Product akan merangkai setiap baris (record) di A dengan setiap di B, sehingga jika A terdiri atas dari n -record dan B terdiri atas m -record, maka jumlah record akan terdiri atas mn record.

4. Penggabungan Himpunan (\cup)

Gabungan (\cup) adalah operasi untuk menghasilkan gabungan dokumen dengan syarat kedua tabel memiliki atribut yang sama, yaitu domain atribut ke - i masing - masing tabel harus sama. Gabungan (union) dari himpunan A dan B adalah himpunan yang setiap anggotanya merupakan anggota himpunan A atau himpunan B.

Operator Union dinyatakan sebagai

$A \cup B = \{X \mid X \in A \text{ or } X \in B\}$,

dimana A dan B haruslah *union-compatible*:

- Memiliki jumlah field yang sama
- Field yang berurutan, dalam urutan kiri ke kanan memiliki domain yang sama

5. Selisih Himpunan ($-$)

Selisih himpunan berfungsi untuk mendapatkan nilai yang ada dalam sebuah tabel relasi, tapi tidak ada dalam tabel relasi lainnya. Simbol " $-$ " digunakan untuk menunjukkan operasi set difference. Selisih dari dua himpunan A dan B adalah suatu himpunan yang elemennya dari A tetapi bukan elemen dari B.

Operator irisan dinyatakan sebagai

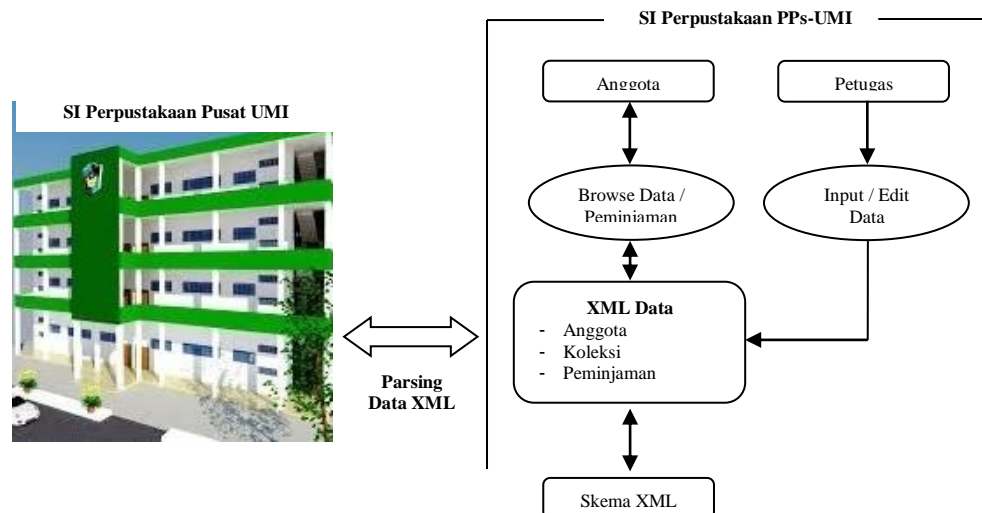
$A - B = \{x \mid x \in A \text{ dan } x \notin B\}$,

dimana relasi A dan B harus union-compatible dan skema hasil identik dengan skema A.

3. Metode Penelitian

3.1. Analisis Sistem

Penelitian ini fokus pada analisis pengoptimalan dalam pencarian data dengan menggunakan operator-operator aljabar relasional dan penerapan skema XML untuk solusi efisiensi dan integrasi data dalam melakukan proses manipulasi informasi pada suatu dokumen. Perancangan sistem yang akan dilakukan seperti pada gambar 1. sebagai berikut :



Gambar 1. Rancangan Umum Sistem

3.2. Tahapan Penelitian

Adapun tahapan-tahapan yang dilakukan dalam penelitian ini yaitu :

- Studi literatur, yaitu dengan melakukan studi dari buku-buku yang berkaitan dengan masalah, juga melalui artikel dari internet.
- Pengambilan sampel, yaitu penentuan lokasi penelitian untuk mendapatkan hasil yang maksimal.
- Pengolahan sampel, yaitu menentukan kebutuhan Sistem Informasi Perpustakaan pada Program Pascasarjana UMI.
- Desain sistem, yaitu merupakan tahap merancang sistem yang akan dibangun.
- Coding / implementasi adalah tahap mengimplementasikan rancangan sistem ke dalam kode program.
- Pengujian sistem merupakan tahap simulasi terhadap sistem yang telah dibuat serta menguji apakah sistem telah sesuai dengan tujuan yang diinginkan atau masih perlu diperbaiki.

3.3. Analisis Kebutuhan

Analisis kebutuhan perangkat yang digunakan dalam penelitian terdiri atas perangkat keras (*hardware*) laptop atau personal computer dan perangkat lunak (*software*) berupa Sistem Operasi Linux dengan Distribusi Ubuntu, XAMPP, dan Notepad++.

4. Hasil dan Pembahasan

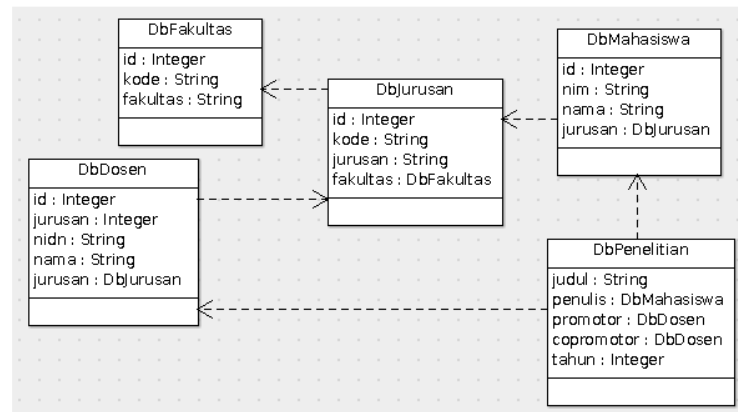
Penelitian ini menghasilkan sebuah *library* yang dapat digunakan untuk mengakses data dalam format XML. *Library* ini dibuat menggunakan bahasa pemrograman C/C++ dan didesain untuk digunakan sebagai *extension* pada pemrograman web PHP. Operasi yang dapat dilakukan mencakup penulisan (*insert*) data, perubahan (*update*) data, hapus (*delete*) data dan pengambilan (*select*) data.

Library ini diterapkan pada kasus Sistem Informasi Perpustakaan dengan menggunakan lima tabel, yakni *tabel mahasiswa*, *tabel dosen*, *tabel fakultas*, *tabel jurusan* dan *tabel penelitian*. Kelima tabel ini diformat kedalam ekstensi XML. Kelima tabel tersebut saling berelasi satu sama lain dan dilakukan beberapa normalisasi agar dapat mengurangi redundansi data. Bahasan ini akan dicakup dalam desain sistem penulisan ini.

4.1. Desain Sistem

1. Normalisasi Dan Relasi Antar Tabel

Tabel fakultas, tabel dosen dan tabel mahasiswa tidak mengalami normalisasi karena merupakan tabel master yang menjadi referensi tabel jurusan dan tabel penelitian. Hasil relasi dari keseluruhan tabel tersebut dapat dilihat pada gambar 2.



Gambar 2. Relasi Antar Tabel

2. Struktur XML

Seluruh data yang digunakan disimpan kedalam format XML yang merepresentasikan kelima tabel yang digunakan. Terdapat beberapa *tag* penting yang menjadi pemisah antar tabel-tabel tersebut. Setiap tag merupakan satu node dalam XML dan setiap kelompok data yang tersimpan adalah sebuah *node*, sedangkan untuk setiap kolom (field) dalam tabel direpresentasikan kedalam *attribute-node*. Gambar 3. menyajikan struktur skema XML untuk setiap tabel.

```
<?xml version="1.0"?>
<db name="dblibumi">

<tb name="tbdosen" key="id" unique="nidn">
<dosen id="id" nidn="nidn" nama="nama dosen" jurusan="0" />

<tb name="tbmahasiswa" key="id" unique="nim">
<mahasiswa id="id" nim="nim" nama="nama mahasiswa"
jurusan="0" />
</tb>

<tb name="tbfakultas" key="id">
<fakultas id="0" nama="Teknik" />
</tb>

<tb name="tbtesis" key="id">
<tesis judul="judul tesis" tahun="2015" penulis="id
mahasiswa" promotor="id dosen" copromotor="id dosen" />
</tb>

<tb name="tbjurusan" key="kode">
<jurusan id="0" nama="Teknik Informatika" fakultas="0"
kode="IT" />
</tb>
</db>
```

Gambar 3. Struktur Skema XML

4.2. Analisis Sistem

1. Operasi Penambahan

Pada sistem database konvensional, seperti SQL, operasi penambahan data dilakukan dengan memberikan perintah *insert* yang disertai dengan beberapa parameter. Operasi penyimpanan data dalam penelitian ini menggunakan pendekatan yang sama dengan system *database* konvensional, namun dibuat kedalam bentuk prosedur. Adapun contoh pemanggilannya di dalam kode PHP adalah sebagai berikut :

```
<?php $hasil = insert("nama_tabel","field1","field
2","..."field n");
```

Adapun variabel \$hasil merupakan nilai balik yang dihasilkan oleh, operasi *insert*. Nilai balik ini bertipe data *integer* dan masing-masing nilainya mendeskripsikan sukses tidaknya proses yang dilakukan. Masing-masing nilai balik ini dapat dilihat pada tabel 1.

Tabel 1. Keterangan nilai balik

Nilai Balik	Keterangan
1	Operasi berhasil dilakukan, file XML ter-update.
0	Tidak terdapat <i>error</i> , operasi berhasil namun kriteria yang dicari tidak ditemukan, sehingga tidak ada perubahan di dalam pusat data
404	File XML tidak ditemukan.
500	File XML ditemukan, namun dengan parameter nama tabel salah

Berikut beberapa penerapan yang diterapkan ke dalam studi kasus sistem perpustakaan.

```
<?php $hasil = insert_academician ("tbmahasiswa",  
"1112.2014.007", "Foo Bar", "IT");
```

Baris kode ini akan membentuk satu *child node* pada file XML pada tabel mahasiswa. Adapun hasilnya adalah sebagai berikut:

```
<?xml version="1.0">  
<db name="dblibrary">  
<tb name="tbjurusan">  
<jurusan id="0" kode="IT"  
nama="Teknik Informatika" />  
</tb>  
<tb name="tbmahasiswa">  
<mahasiswa id="n" nim="1112.2014.007"  
nama="Foo Bar" jurusan="0" />  
</tb>  
</db>
```

2. Operasi Hapus

Sama seperti operasi penambahan data, yang menggunakan pendekatan database konvensional. Jika dalam bahasa query implementasi dilakukan dengan perintah:

```
"delete from nama_tabel where nama_field = nilai"
```

Maka dalam penelitian ini kami mengimplementasikannya sebagai berikut:

```
<?php $hasil = delete_from("nama_tabel",  
"field_kriteria_pencarian", "nilai");
```

Dengan memanggil perintah ini, maka sistem akan mencari didalam pusat data satu persatu. Jika nilai yang diberikan sebagai parameter ditemukan dalam tabel, maka sistem akan mengembalikan nilai satu, sedangkan jika nilai tidak ditemukan maka akan mengembalikan nilai nol.

3. Operasi Pencarian

Seperti halnya dengan RDMS konvensional yang menggunakan bahasa SQL dalam pencariannya, penelitian ini juga menggunakan pendekatan yang sama tetapi dengan menggunakan XQUERY sebagai modul pencariannya. XQUERY ditulis kedalam sebuah file, dan menjadi parameter dalam prosedur pemanggilannya.

3.1. Implementasi Selection

Proses *selection* diterapkan dengan menggunakan perintah "*for ... in*" dengan diikuti parameter file XML serta nama *tag* yang akan menjadi *query*. Adapun *query*-nya dapat dituliskan dengan:

```
for $var in  
doc("file.xml")/db/tb[@name='nama_tabel']/tag_tabel  
where kondisi_selection
```

Keterangan:

- *\$var*, merupakan alias dari nama tabel yang akan di-*query*.
- **File.xml**, merupakan nama file XML yang akan digunakan sebagai sumber data.
- **/db/tb**, berurut masing-masing merupakan *tag* <db> dan <tb> dalam file XML.
- **[@name='nama_tabel']**, diisi dengan nama tabel yang akan diquery
- **tag_label**, merupakan child node dari *nama_tabel*.
- **where kondisi_selection**, akan diisi dengan nilai kondisi *query* sebagai *filter* dari hasil.

Misalnya hendak dilakukan pencarian nama mahasiswa, dilakukan dengan:

```
for $mhs in
doc("umi.xml")
/db/tb[@name='tbmahasiswa']/mahasiswa
where $mhs/string(@nim) = "1134.421.33"
return $mhs
```

Query ini akan memberikan nilai balik dengan hasil berupa data mahasiswa dengan nilai *nim* sama dengan 1134.421.33, selain itu akan diabaikan.

3.2. Implementasi Projection

Sama halnya dengan proses *selection*, proses *projection* juga dilakukan dengan formula query yang sama tetapi dengan nilai hasil yang lebih spesifik. *Query* dapat menentukan kolom / *field* apa saja yang hendak kita jadikan sebagai nilai balik. *Query projection* dalam sistem ini dilakukan dengan:

```
for $mhs in
doc("umi.xml")
/db/tb[@name='tbmahasiswa']/mahasiswa
where $mhs/string(@nim) = "1134.421.33"
return { $mhs/string(@nim) } { $mhs/string(@nama) }
```

Dengan penambahan *projection* {*\$mhs/string(@nim)*} dan {*\$mhs/string(@nama)*} nilai hasil yang diberikan hanya sebatas kolom *nim* dan *nama*.

3.3. Implementasi Join

Seperti halnya dengan dua proses sebelumnya, operasi *join* juga menggunakan perintah "*for .. in*". Perbedaannya adalah dengan mengimplementasikan *join* tabel yang memiliki relasi satu sama lain dapat digabungkan menjadi satu *view*. Untuk dapat menghasilkan *output* seperti yang terlihat pada gambar 4.1, query dapat dilakukan dengan perintah:

```
for
$mhs in
doc("umi.xml")/db/tb[@name='tbmahasiswa']/mahasiswa,
$jrs in
doc("umi.xml")/db/tb[@name='tbjurusan']/jurusan[@kode=
$mhs/@jurusan],
$tss in
doc("umi.xml")/db/tb[@name='tbtesis']/tesis[@penulis=$
mhs/@id],
$pro in
doc("umi.xml")/db/tb[@name='tbdosen']/dosen[@id=$tss/@
promotor],
$cop in
doc("umi.xml")/db/tb[@name='tbdosen']/dosen[@id=$tss/@
copromotor],
$fkt in
doc("umi.xml")/db/tb[@name='tbfakultas']/fakultas[@id=
$jrs/@fakultas]
return
{ $mhs/string(@nim) }
```

```
{ $mhs/string(@nama) }  
{ $fkt/string(@nama) }  
{ $jrs/string(@nama) }  
{ $pro/string(@nama) }  
{ $scop/string(@nama) }  
{ $stss/string(@judul) }  
{ $stss/string(@tahun) }
```

4.3. Uji Coba Program

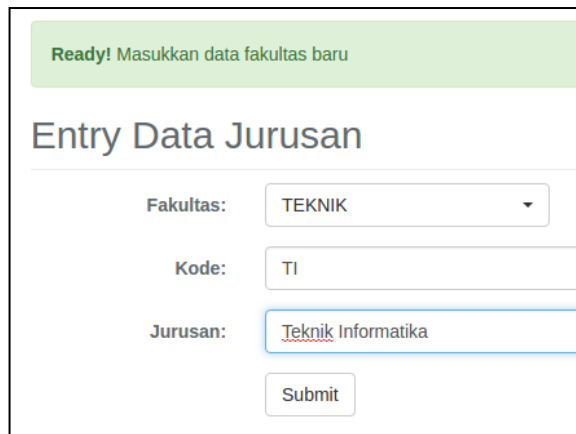
Program diuji coba dengan menggunakan sistem dashboard pada dashboard sistem perpustakaan. Adapun operasi yang diujikan yaitu operasi penginputan data dan operasi pencarian data.

1. Penginputan Data

Pengujian penginputan data dilakukan dengan memasukkan data kedalam tabel mahasiswa, tabel dosen, tabel dosen, tabel jurusan, tabel fakultas dan tabel penelitian.

Pengujian penginputan tabel jurusan dilakukan dengan mengisi kolom *kode jurusan*, *kode fakultas* dan *nama jurusan*. Adapun ketiga kolom ini bertipe data string serta terdapat satu kolom *id* dengan metode *auto-increment*.

Uji coba berikut, dilakukan dengan mengisikan *string* "IT" kedalam kode jurusan, *string* "TK" kedalam kode fakultas dan *string* "Teknik Informatika" sebagai nama jurusan. Berikut tampilan implementasinya:



Gambar 4. Tampilan Input Data Jurusan

Dengan melakukan *submit*, seluruh data akan diproses dan akan disimpan kedalam database XML. Dengan demikian, data yang berada didalam database XML akan terupdate seperti yang terlihat pada gambar 5.

```
<tb name="tbjurusan" key="kode">  
  <jurusan id="0" fakultas="0" kode="TG" nama="Teknik Geologi"/>  
  <jurusan id="1" fakultas="0" kode="TE" nama="Tekni Elektro"/>  
  <jurusan id="2" fakultas="0" kode="TM" nama="Teknik Mesin"/>  
  <jurusan id="3" fakultas="0" kode="AR" nama="Teknik Arsitektur"/>  
  <jurusan id="4" fakultas="0" kode="SP" nama="Teknik Sipil"/>  
  <jurusan id="5" fakultas="1" kode="IE" nama="Ilmu Ekonomi"/>  
  <jurusan id="6" fakultas="1" kode="MM" nama="Ekonomi Manajemen"/>  
  <jurusan id="7" fakultas="2" kode="MT" nama="Ilmu Matematika"/>  
  <jurusan id="8" fakultas="2" kode="CS" nama="Ilmu Komputer"/>  
  <jurusan id="9" fakultas="2" kode="PH" nama="Ilmu Fisika"/>  
  <jurusan id="10" fakultas="2" kode="BH" nama="Ilmu Biologi"/>  
  <jurusan id="11" fakultas="0" nama="Teknik Informatika" kode="TI"/>  
</tb>
```

Gambar 5. Member Tabel Jurusan Dengan Data Baru

2. Pencarian Data

Pengujian penginputan data dilakukan dengan memasukkan data kedalam form dan membaginya kedalam kriteria pencarian, seperti: pencarian berdasarkan judul penelitian, pembimbing atau nama penulis. Pencarian disini dirangkumkan kedalam tabel penelitian, dimana tabel penelitian terelasi dengan seluruh tabel yang ada dalam database. Struktur relasi untuk setiap tabel dapat dilihat pada gambar 2 dalam tulisan ini. Adapun tampilan pencariannya dapat dilihat pada gambar 6.

Cari Berdasarkan:

Keyword:

Daftar Penelitian

#	NIM	MAHASISWA	FAKULTAS	JURUSAN	PROMOTOR	COPROMOTOR	JUDUL	TAHUN
0	0155.01.39.2013	Hamka	EKONOMI	Ekonomi Manajemen	Prof.Dr.Hj.Masdar Masud,SE.,M.Si	Dr.Hj.Nurpadila,SE.,MM	ANALISIS KINERJA PENGELOLAAN KEUANGAN DAERAH PEMERINTAH KABUPATEN MAROS TAHUN ANGGARAN 2010-2014	2015

Gambar 6. Tampilan Pencarian Data

Dengan memberikan parameter kategori dan *keyword* pencarian dijalankan dengan mem-*filter* pencarian sesuai dengan kriteria pencariannya. Dalam bahasa *query* pencarian ini dibuat dengan perintah SELECT JOIN, sedangkan dalam proses menggunakan struktur XML, perintah yang kami gunakan adalah sebagai berikut:

```
for
$mhs in
doc("../xml/umi.xml")/db/tb[@name='tbmahasiswa']/mahasiswa,
$jrs in
doc("../xml/umi.xml")/db/tb[@name='tbjurusan']/jurusan[@id=$mhs/@jurusan],
$stss in
doc("../xml/umi.xml")/db/tb[@name='tbtesis']/tesis[@penulis=$mhs/@id],
$pro in
doc("../xml/umi.xml")/db/tb[@name='tbdosen']/dosen[@id=$stss/@promotor],
$scop in
doc("../xml/umi.xml")/db/tb[@name='tbdosen']/dosen[@id=$stss/@copromotor],
$fkt in
doc("../xml/umi.xml")/db/tb[@name='tbfakultas']/fakultas[@id=$jrs/@fakultas]
return
<tr>
  <td>{$mhs/string(@id)}</td>
  <td>{$mhs/string(@nim)}</td>
  <td>{$mhs/string(@nama)}</td>
  <td>{$fkt/string(@nama)}</td>
  <td>{$jrs/string(@nama)}</td>
  <td>{$pro/string(@nama)}</td>
  <td>{$scop/string(@nama)}</td>
  <td>{$stss/string(@judul)}</td>
  <td>{$stss/string(@tahun)}</td>
</tr>
```

Keuntungan dengan menggunakan query ini yakni nilai balik dapat diformat sesuai dengan model tabel yang kita kehendaki. Dalam hal ini, nilai balik yang dihasilkan berupa struktur tabel dalam format HTML. Secara visual relasi yang dibentuk dari query ini dapat dilihat pada gambar 2.

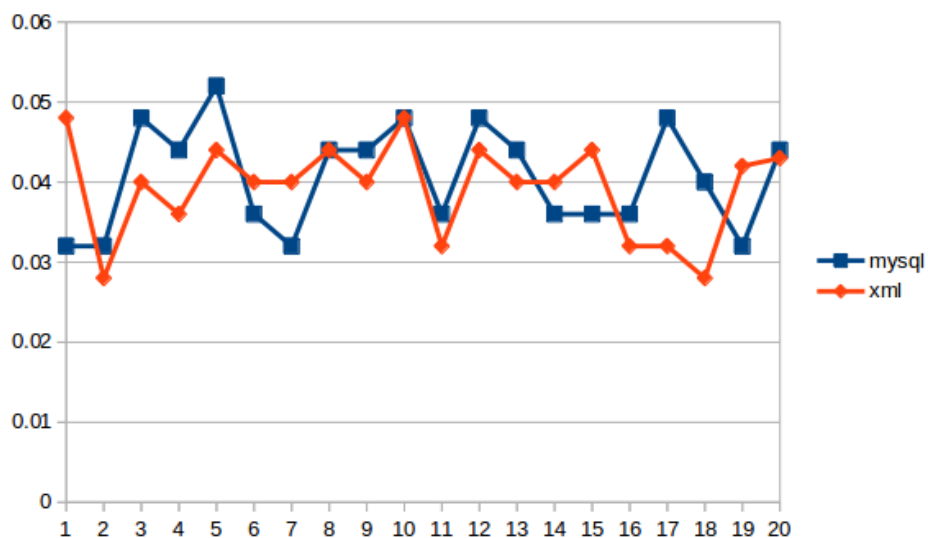
4.4. Analisis Waktu Eksekusi

Hasil Percobaan dan Waktu Eksekusi Dari Kinerja Sistem Dapat dilihat sebagai berikut :

Tabel 2. Hasil Percobaan dan Waktu Eksekusi

Percobaan ke-	MySQL (detik)	XML (detik)
1	0,0320	0,0480
2	0,0320	0,0280
3	0,0480	0,0400
4	0,0440	0,0360
5	0,0520	0,0440
6	0,0360	0,0400
7	0,0320	0,0400
8	0,0440	0,0440
9	0,0440	0,0400
10	0,0480	0,0480
11	0,0360	0,0320
12	0,0480	0,0440
13	0,0440	0,0400
14	0,0360	0,0400
15	0,0360	0,0440
16	0,0360	0,0320
17	0,0480	0,0320
18	0,0400	0,0280
19	0,0320	0,0420
20	0,0440	0,0430
Rata-Rata	0,0406	0,0393

Waktu eksekusi yang dihitung dalam penelitian ini adalah waktu eksekusi pencarian *query* data dengan membandingkan antara waktu program yang dijalankan dengan data pusat *mysql* dengan sistem yang dibuat kedalam XML. Sehingga sebagai pembandingan dibuatkan sebuah data pusat *mysql* dengan struktur yang menyerupai relasi XML yang digunakan. Adapun perbandingannya dapat dilihat pada tabel 2.



Gambar 7. Grafik Perbandingan

Dari hasil percobaan yang dilakukan, rata-rata waktu *query* pencarian data menggunakan *MySQL* adalah sebesar 0.0406 detik sedangkan waktu pencarian *query* menggunakan *XML* adalah sebesar 0.0393 detik. Ini berarti dengan menggunakan XML, dapat mengurangi waktu eksekusi hingga 1,03 kali dibandingkan dengan database client server seperti *mysql*.



5. Kesimpulan

Penelitian ini menunjukkan bahwa, operasional aljabar relasional pada sistem data dapat diterapkan. Data yang telah disimpan kedalam format XML dapat diakses menggunakan operasi relasional, terutama untuk operasi pencarian yang merupakan inti dari sebuah aplikasi *database*. Sistem yang dibuat dapat digunakan pada aplikasi dengan jumlah relasi data yang sederhana dan menghemat waktu operasional dibandingkan dengan sistem database yang konvensional yang dalam penelitian dapat lebih efisien sebesar 1,03 kali. Sistem dengan database, SQL misalnya, dapat diintegrasikan dengan *me-load* data yang sering digunakan kedalam format XML. Dengan demikian waktu komunikasi antara client dan server database dapat dikurangi.

Referensi :

- [1] Undang – Undang RI. No. 43 Tahun 2007 tentang perpustakaan.
- [2] A. Ibrahim and M. Afrina, "Pengembangan Model Perpustakaan Berbasis Teknologi Informasi untuk meningkatkan kinerja layanan perpustakaan dan mewujudkan perpustakaan ideal berbasis digital di Fasilkom Unsri," in *Prosiding Konferensi Nasional Teknologi Informasi dan Aplikasinya*, Palembang, 2011.
- [3] Manes, A.T., "Introduction to Webservices", 2001.
- [4] Hendra Kusmayadi, ED, "*XML and Web Service*". Bandung : Polytechnic Telkom Bandung, 2009. (in Indonesia).
- [5] Thorsten Fiebig, J. W. RAW : A. Relational Algebra For The Web. *Journal of Fakulty Fur Matematik And Informatik*, 1997.
- [6] Aji Supriyanto. 2007. Web dengan HTML dan XML. Graha Ilmu. Yogyakarta.
- [7] Fuhr N, Rolleke T. A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *ACM Transactions on Information Systems*. 1997; XV(1): p.32-66.
- [8] Silberschats A, Korth HF, Sudarshan S. Database System Concept. In. International : McGraw-Hill; 2010. p. 218.

