

PENERAPAN TEXTUAL HEIARCHY (PATHS) UNTUK PERMASALAHAN AKSES DATA NILAI AKADEMIK MAHASISWA

Afriyudi

Sistem Informasi Universitas BINA DARMA Palembang
Jl Jend. A. Yani No.12, Palembang, Sumatera Selatan 30264
Email : babeyudi@mail.binadarma.ac.id , babeyudi@gmail.com

Abstrak

Data nilai mahasiswa bagi perguruan tinggi, merupakan data yang sangat bernilai. Perguruan tinggi yang menerapkan komputerisasi, menyimpan Data nilai dalam format elektronik yang dikenal dengan nama database. Database relational merupakan pilihan utama untuk menyimpan data nilai tersebut. ketika jumlah data nilai mahasiswa masih sedikit proses pengaksesan dapat dilakukan dengan cepat. Bertambahnya jumlah mahasiswa dari tahun-ketahun tentunya akan menyebabkan ukuran data nilai mahasiswa menjadi besar sehingga permasalahan pengaksesan biasanya akan terjadi. pada penelitian ini akan di terapkan metode yang konvensional yaitu menerapkan metode textual hierarchy (path) yang umumnya digunakan untuk hierarchical data model untuk permasalahan data relational. Merupakan hal yang sangat menarik menerapkan salah satu metode hierarchical data model pada model data relational. hasil yang diharapkan adalah dapat mengetahui sejauh mana metode textual hierarchy (path Enumeration) dapat membantu mempercepat pengaksesan data relational dalam ukuran yang besar.

Kata kunci: *textual heierarchy, data relational, nilai akademik.*

1. Pendahuluan

Dewasa ini penggunaan Aplikasi komputer untuk pengelolaan data nilai akademik merupakan suatu yang mutlak harus dilakukan oleh sebuah lembaga pendidikan. Database merupakan teknologi penyimpanan data yang dapat digunakan dalam menyimpan data nilai akademik. Selain dari pada itu, Penggunaan database ini memungkinkan kumpulan data yang telah disimpan tersebut dapat di kelola secara sistematis seperti pembuatan laporan, backup data dan lain-lain. Dalam pengelolaannya di kenal beberapa macam model database yaitu relational, network, hierarchy dan lain-lain. Umumnya teknologi yang digunakan adalah relational data model, model ini menggambarkan semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model

ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel.

Relational data model merupakan pilihan utama untuk mengelola data nilai akademik. Kemudahan pengelolaan merupakan faktor utama yang menyebabkan model ini dipilih. Kenyataannya penggunaan model ini juga memiliki keuntungan dan kerugian. Kerugiannya adalah jika model yang dikelola menjadi kompleks dan intensitas data meningkat, maka pengaksesan data menjadi lebih lambat. Kompleksitas terjadi karena data nilai merupakan hasil relasi dari beberapa tabel data yaitu tabel MataKuliah, Dosen, KartuRencanaStudi, Mahasiswa, ProgramStudi, Fakultas dan jadwal. Permasalahan terjadi ketika data nilai ini digunakan untuk membuat transkrip nilai, ditambah lagi intensitas data setiap semester semakin bertambah dan data nilai setiap fakultas di simpan dalam satu tabel data nilai. Kompleksitas dari relasi yang digunakan membuat pengaksesan data menjadi lambat.

Hierarchical data model merupakan salah satu alternatif yang dapat digunakan untuk mengatasi permasalahan ini. hierarchical data model adalah model data di mana data diorganisasi dalam struktur seperti pohon. Struktur ini memungkinkan pengulangan informasi menggunakan konsep hubungan parent dan child dimana setiap parent dapat memiliki banyak child tetapi setiap child hanya memiliki satu parent (dikenal sebagai 1:Many dan 1:1)[1]. Ada 4 macam metode yang dapat digunakan untuk mengakses Hierarchical data model yaitu Recursion, stack, Flat table dan Modified Preorder Tree Traversal Algorithm [6]. Sedangkan menurut Tulder , Ada dua algoritma yang dapat digunakan untuk storing data Hierarchical yaitu The Adjacency List Model atau recursion method dan Modified Preorder Tree Traversal [5]. Bill Karwin dalam presentasinya menjelaskan bahwa ada 4 macam metode yang dapat digunakan untuk store dan query pada Hierarchical data yaitu Adjacency list , Path enumeration , Nested sets dan Closure table .[2]. Ada dua macam metode yang dapat digunakan untuk model path Enumeration yaitu edge enumeration dan node enumeration[4]. Pada edge enumeration model sebagai panduan untuk mengakses data mulai dari awal sampai akhir menggunakan notasi angka integer untuk pointer, sedangkan node enumeration menggunakan notasi string sebagai panduan untuk mengakses data dari awal sampai akhir.[3]

Sifat data akademik potensial untuk dapat dibuat menjadi tree(one to many). Dasar permasalahan inilah yang menyebabkan peneliti tertarik menerapkan algoritma Textual Heirarchy (path Enumeration) untuk menyelesaikan permasalahan tersebut, metode yang digunakan adalah edge enumeration dan pada penelitian ini akan dilakukan perubahan algoritma pada metode tersebut.

2. Perancangan dan Implementasi

2.1 Perancangan tabel

Sesuai dengan teori dalam pembuatan tabel dengan menerapkan edge enumeration. Untuk menerapkan metode ini akan ada dua tabel yang tercipta yaitu tabel data dan tabel path. Tabel data merupakan tabel yang berisi data nilai mahasiswa dan tabel path merupakan tabel yang berisi path atau jalur (merupakan kumpulan nomor record) dengan menggunakan metode edge enumeration. Dasar pemikiran untuk merubah algoritma dasar edge enumeration adalah dengan penggunaan perintah IN pada perintah SQL, sehingga path yang tercipta pada metode ini adalah alamat record dengan format tertentu yang dapat digunakan dengan perintah IN pada SQL. Ada dua tabel yang dihasilkan dari tahapan ini yaitu tabel nilai_detail dan tabel nilai_rekap, Tabel nilai_detail merupakan data yang berisi nilai-nilai hasil studi seluruh mahasiswa dan tabel nilai_rekap merupakan kumpulan jalur atau path pada tabel nilai yang merupakan kumpulan alamat record untuk masing-masing data nilai mahasiswa dengan menggunakan pola paterrn koma (,) untuk memisahkan masing-masing nomor record. Berikut ini merupakan rancangan tabelnya.

Tabel 1 : tabel nilai_detail

No	Nama Field	Type	Length	Keterangan
1	Nim	Varchar	9	Nim mahasiswa
2	th_akademik	Varchar	5	Tahun akademik
3	Sms	Varchar	1	1 = Genap 0 = Ganjil
4	kd_matkul	Varchar	11	Kode mata kuliah
5	Smt	Varchar	1	Semester
6	Nilai	enum('A', 'B', 'C', 'D', 'E')		Nilai
7	Recid	Int		No record yang digenerate secara otomatis

Tabel 1 merupakan tabel yang berisikan data nilai mahasiswa, sesuai dengan petunjuk pembuatan tabel dengan metode hierarchical, harus ada record penanda yang merupakan pointer yang dapat digunakan untuk mengakses data tersebut. Field recid, merupakan pointer yang digunakan untuk penanda. Field ini nantinya digunakan sebagai path atau jalur yang dirangkum dalam tabel nilai rekap. Untuk memudahkan penomoran, field ini dibuat dengan tipe data interger dan dibuat auto number, sehingga field ini nantinya secara otomatis akan diberi nomor record secara otomatis dan terurut.

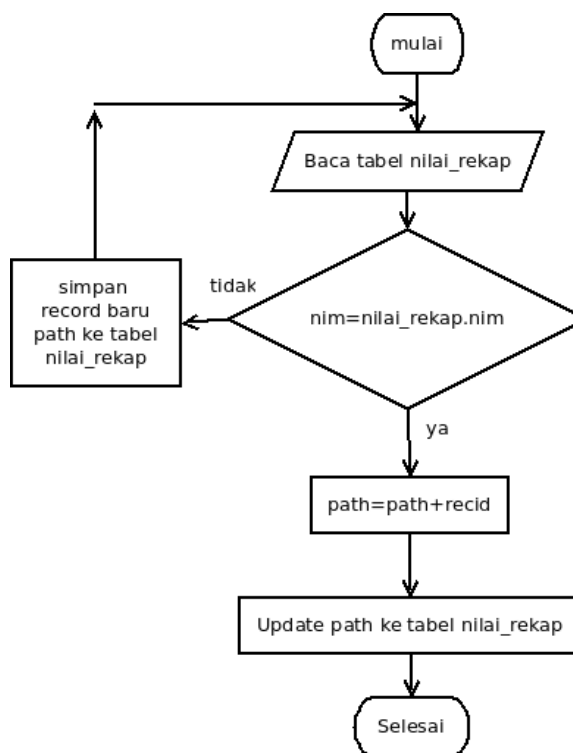
Tabel 2 : tabel nilai_rekap

No	nama Field	Type	panjang	Keterangan
1	Nim	varchar	9	Nim mahasiswa
2	path	text		Alamat jalur/path

Tabel 2 merupakan tabel yang berisikan path atau jalur dari hasil rekapitulasi nomor record pada tabel detail nilai.

2.2 Algoritma Penyimpanan path enumeration

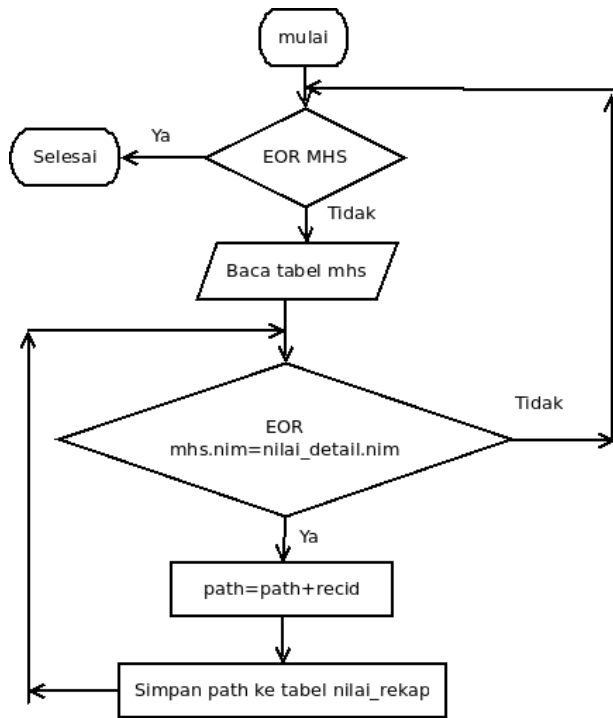
Ada dua algoritma yang dapat digunakan untuk menerapkan model penyimpanan data nilai_rekap, yaitu algoritma penyimpanan untuk pertama kali dan Penyimpanan untuk record baru.



Gambar 1. Algoritma menyimpan path lama

Penjelasan algoritma Gambar 1 :

1. Membaca nim di tabel nilai rekap,
2. jika ada maka tambahkan nomor record pada path dan simpan pada tabel nilai rekap.
3. Jika nim tidak ada maka simpan nim dalam tabel nilai rekap.



Gambar 2. Algoritma menyimpan path baru

Berikut ini merupakan penjelasan algoritma pada Gambar 2, algoritma ini dipakai pertama kali pada saat tabel nilai_rekap dibuat untuk pertama kali.

Penjelasan algoritma Gambar 2 :

1. Membaca seluruh nim di tabel MHS,
2. Membaca tabel nilai_detail berdasarkan nim pada tabel MHS.
3. Simpan recid pada tabel nilai_rekap jika seluruh nilai nim tertentu telah habis dibaca, kemudian dilanjutkan ke langkah 2 kembali
4. Jika record pada tabel MHS telah habis maka selesai.

Source Program Algoritma menyimpan path lama:

```

$query_data_krs = "SELECT recid from
nilai_detail where nim like '$nimnya'
order by recid";

$data_krs
mysql_query($query_data_krs, $babe11)
or die(mysql_error());
$row_data_krs
mysql_fetch_assoc($data_krs);
    
```

```

$lok=" ";
do
{
    $lok.=$row_data_krs['recid'].",";
} while($row_data_krs =
mysql_fetch_assoc($data_krs));
$lokasi=substr($lok,0,strlen($lok)-
1);
mysql_free_result($data_krs);
mysql_select_db($database_babe11,
$babe11);
$query_sudah_nilai_rekap = "select *
from nilai_rekap where nim='$nimnya'";

$sudah_nilai_rekap
mysql_query($query_sudah_nilai_rekap,
$babe11) or die(mysql_error());
$totalRows_data
mysql_num_rows($sudah_nilai_rekap);
if($totalRows_data==0) // jika path
kosong
{
    mysql_select_db($database_babe11,
$babe11);
$query_data_nilai_rekap = "insert
into nilai_rekap(nim,path)
values('$nimnya', '$lokasi')";
$simpan_nilai_rekap
mysql_query($query_data_nilai_rekap,
$babe11) or die(mysql_error());
}
else
{
    mysql_select_db($database_babe11,
$babe11);
$query_data_nilai_rekap = "update
nilai_rekap set data='$lokasi' where
nim='$nimnya'";
$simpan_nilai_rekap
mysql_query($query_data_nilai_rekap,
$babe11) or die(mysql_error());
}
}
    
```

Source Program Algoritma menyimpan path baru:

```

transferRecidRecord()
{
    try {
        Class.forName("org.gjt.mm.mysql.Driver
");
    }
    catch(java.lang.ClassNotFoundException
e)
    {
        System.err.print("ClassNotFoun
dException: ");
        System.err.println(e.getMes
sage());
    }

    try {
        Connection connection =
        DriverManager.getConnection("jdb
c:mysql://localhost/mhsbidar1?user=ro
ot&password=");
    }
}
    
```

```

        Statement      statement      =
connection.createStatement();

        // membaca seluruh nim mahasiswa
        ResultSet      rs            =
statement.executeQuery("select nim from
mhs");

        int nomor=0;
        while(rs.next())
        {
            nomor++;
            String      vnim          =
rs.getString(1);
            System.out.println("No
"+nomor);

            Statement  statement1    =
connection.createStatement();

            // membaca detail alamat record
masing-masing mahasiswa
            ResultSet  rs1           =
statement1.executeQuery("select      recid
from      detail_nilai      where
nim='"+vnim+"'");

            String path="";
            while(rs1.next())
            {
                int vnorec = rs1.getInt(1);
                path=path+Integer.toString(vnorec)+",";
            }
            String vpath="";
            if(path.length(>1)

vpath=path.substring(0,path.length()-
1);

            else
                vpath=path;
            statement1.close();

            Statement  statement2    =
connection.createStatement();

            // Simpan detail alamat
record/path ke dalam tabel nilai rekap
            int      rs2            =
statement2.executeUpdate("insert into
nilai_rekap(nim,path)
values('"+vnim+"','"+vpath+"')");
        }
        statement.close();
        connection.close();
    }
    catch(SQLException ex)
    {
        System.err.println("SQLException: " +
ex.getMessage());
    }
}
    
```

2.3 Pengaksesan data

Pada tahap ini akan dicoba melakukan query untuk membaca data nilai mahasiswa dengan algoritma yang telah dijabarkan sebelumnya.

Query membaca path :

```

$query_data_mhs = "select * from
nilai_rekap where nim='$nim'";
$data_mhs = mysql_query($query_data_mhs,
$babel) or die(mysql_error());
$row_data_mhs =
mysql_fetch_assoc($data_mhs);
$datanya=$row_data_mhs['data'];
    
```

Query membaca data nilai :

```

$query_baca_smt = "SELECT
nim,th_akademik,sms,kd_matkul, nilai FROM
nilai_detail where nilai_detail.recid
in('.$datanya.') group by th_akdemik,sms
order by th_akdemik,sms DESC";
$data_baca_smt =
mysql_query($query_baca_smt, $babel) or
die(mysql_error());
    
```

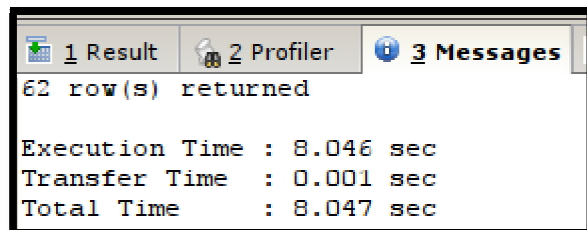
Uji coba dengan jumlah record data = 1.015.897 record, dengan sqlyog untuk menjalankan query.

Hasil query tanpa path :

```

SELECT      nim,th_akademik,sms,kd_matkul,
nilai      FROM      nilai_detail      WHERE
nim='94141001';
    
```

Output :



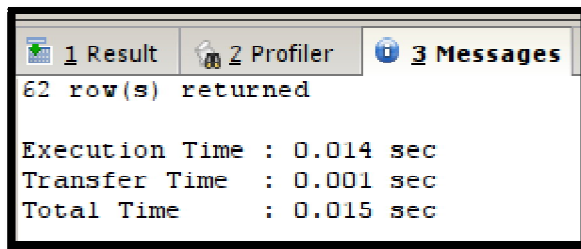
Gambar 3. Hasil query tanpa path

Hasil query dengan path :

```

SELECT      nim,th_akademik,sms,kd_matkul,
nilai FROM nilai_detail WHERE norec IN
(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
,18,19,20,21,22,23,24,25,26,27,28,29,30,31
,32,33,34,35,36,37,38,39,40,41,42,43,44,45
,46,47,48,49,50,51,52,53,54,55,56,57,58,59
,60,61,62)
    
```

Output :



Gambar 4. Hasil query dengan path

3. Kesimpulan

Dari ujicoba yang dilakukan dapat diambil kesimpulan bahwa penggunaan path enumeration meningkatkan waktu akses dalam pembacaan record dengan record yang cukup besar yaitu 1.015.897 record. Kemudian metode path enumeration ini akan lebih meningkat waktu aksesnya jika data yang diakses sebelumnya dilakukan sortir atau pengurutan data. Sedangkan untuk menyimpan path untuk record baru akan lebih baik jika digunakan trigger yaitu pada saat tabel detail di tambahkan record baru, maka secara otomatis trigger akan dipanggil.

Daftar Pustaka

- [1] Bagui, Sikha and Earp, Richard., *Database Design Using Entity-Relationship Diagrams*, Auerbach Publications, 2003.
- [2] Bill Karwin, *Models for Hierarchical Data with SQL and PHP* [Online]. Tersedia: <http://www.slideshare.net/billkarwin/models-for-hierarchical-data>, [Di akses 21 Mei 2013]
- [3] Celko,Joe, *Hierarchical SQL*. [Online]. Tersedia: http://www.onlamp.com/2004/08/05/hierarchical_sql.html, [Diakses :21 Mei 2013].
- [4] Celko,Joe, *Joe Celko's Trees And Hierarchies In Sql For Smarties*, Morgan Kaufmann Publishers, 2004.
- [5] Gijis Van Tulder, *Storing Hierarchical Data in a Database* [Online]. Tersedia: <http://articles.sitepoint.com/article/hierarchical-data-database>, [Diakses : 21 Mei 2013].
- [6] Kirby Fling, *Four ways to work with hierarchical data*. [Online]. Tersedia: <http://www.evolt.org/node/4047>, [Diakses :21 Mei 2013].

Biodata Penulis

Afriyudi, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Manajemen Informatika di STMIK Bina Darma Palembang, lulus tahun 1998. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Fakultas MIPA Universitas Gajah Mada Yogyakarta, lulus tahun 2003. Saat ini menjadi Dosen di Universitas BINA DARMA Palembang.

