

PENERAPAN BASIS DATA TERPARTISI DALAM RANCANG BANGUN SISTEM PENGGAJIAN- PENGUPAHAN DENGAN MODEL JARINGAN CLIENT-SERVER

Yulis Dwi Anto ¹⁾ Januar Wibowo ²⁾ Tegar Heru S. ³⁾

1,2,3) S1/Jurusan Sistem Informasi STIKOM Surabaya.

Email: ¹⁾ rsmrr80@yahoo.com, ²⁾ januar@stikom.edu, ³⁾ tegar@stikom.edu

Abstract ; Managing data transaction of employees such as check clock, wages, salaries, income tax, honorarium, commission, gift will become a serious difficult task if involving large quantity employees. You can imagine how busy you are if you have to take care of about 3000 or more data of employees every month with only Microsoft Excel as your tool, you will have more difficulties if you have to take care of them every day. Definitely you need better solution, and because of that here I presenting to you this wages and salaries management system with client server. The use age of client server database model will make server become center of data transaction and requests which mean make server responses will become more and more slower as numbers of client requests increasing. So system will need better strategy to avoid time consuming process. Because server hold large number of historical data, it would be time consuming if user of system try to querying large sum of data query. One of strategy that will be applied to the server that we hope could become as better strategy is database partitioning. Partitioning if the way to divide large number of data table into smaller segments, so hopefully within smaller data to probe for requested query then server will finish the task with more time efficiently.

Memajemen data karyawan dengan menggunakan aplikasi MS. Excel dari hari ke hari akan dirasakan semakin merepotkan karena banyaknya *sheets* dan *workbooks* yang harus dibuka. Belum lagi konsolidasi data antar *administrator* yang membutuhkan waktu dengan data lebih dari seribu karyawan untuk diolah. Penggunaan basis data dengan model distribusi *client-server* ialah pilihan tepat untuk menjadi solusi praktis dalam pendataan karyawan dengan jumlah yang sedemikian. *Client-server* memiliki keunggulan praktis dalam hal integritas data sehingga ketidakefisienan konsolidasi antar *sheets*, antar *workbooks*, dan antar *administrator* dapat ditinggalkan.

Selain itu *client-server* juga lebih praktis untuk tipe transaksi yang menggunakan *online data sharing* pada model kerja bersamaan/*workgroups*. Keuntungan lainnya dengan penggunaan sistem distribusi data terpusat *client-server* ialah lalu lintas data lebih mudah diawasi, selain itu juga data dapat langsung terkumpul dipusat sehingga bisa langsung diolah menjadi informasi.

LANDASAN TEORI

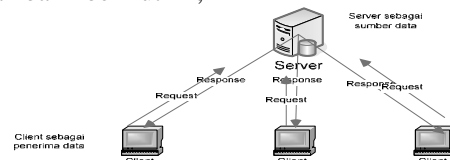
1. Model Jaringan

- **Model Peer to Peer.**

Yaitu model jaringan dimana setiap unit pada jaringan memiliki level yang sama dalam hierarki jaringan. Pada model jaringan ini aliran data bersifat sejajar antar satu unit dengan unit lainnya dalam jaringan.

- **Model Client-Server.**

Merupakan model aliran data yang memiliki baik hierarki maupun hubungan langsung antar komputer dalam jaringan. Hubungan aliran data pada model ini dilakukan secara langsung dari klien ke server tanpa harus melalui protokol ataupun lapisan aplikasi khusus untuk komunikasi antara klien dan server. Model aliran *Client-Server* tergambar pada gambar 1 berikut ini;



Gambar 1. Model Client-Server.

Pada gambar 1 tampak komputer klien berfungsi melakukan *request* dan komputer server melakukan *response* dari *request-request* komputer klien.

- **Model Hierachical Client-Server.**

Merupakan model aliran data yang memiliki karakteristik *Client-Server* tetapi dengan banyak lapisan/*layers*, atau *tiers* antara server-servernya.

- **Model Distributed Computing,**

Merupakan model distribusi data yang memiliki karakteristik sebagai kebalikan data model *Client-Server*. Pada model ini proses *request* tidak dijalankan oleh komputer *cabang/node* tetapi oleh komputer server, yang bertindak sebagai *task manager*.

2. Transaksional Proses Sistem

Online transaction processing atau yang biasa di sebut OLTP akan menuntun para perancang arsitektur basis data pada efisiensi dan ketepatan manajemen penyimpanan data transaksi. Sehingga karakteristik perancangan sistem OLTP ialah; Obyektivitas pencatatan data, Normalitas desain tabel basis data, Kueri yang bersifat spesifik, Akuntabilitas dan realibilitas data, Performa sistem. Karena desain OLTP yang tidak menghendaki kueri data dalam jumlah yang besar maka pemartisian basis data selayaknya tidak diterapkan pada OLTP, Sebagaimana pernyataan dari Thomas Kyte(2010:565) yang merujuk bahwa pemartisian basis data seharusnya tidak diterapkan pada sistem OLTP karena keuntungan-keuntungan dari pemartisian basis data tidak akan berdampak banyak.

3. Pengertian dari Pemartisian Basis Data

Pengertian pemartisian basis data menurut MySQL ialah :

“*Partitioning* takes this notion a step further, by enabling you to distribute portions of individual tables across a *file* system according to rules which you can set largely as needed. In effect, different portions of a table are stored as separate tables in different locations.”

(MySQL,2011:<http://dev.mysql.com/doc/refman/5.1/en/partitioning-overview.html>).

4. Tipe-tipe dari Pemartisian Basis Data

MySQL(2011:<http://dev.mysql.com/doc/refman/5.1/en/partitioning-types.html>) dapat diketahui bahwa pemartisian basis data dapat dilakukan dengan beberapa cara antara lain;

- **Range Partitioning , atau pemartisian berjarak.**

Merupakan bentuk pemartisian sedemikian rupa sehingga nilai ekspresi pada pemartisian terletak antara jarak yang ditentukan. Jarak yang ditentukan haruslah *feasible* dan tidak *overlapping*. Contoh :

```
CREATE TABLE employees (  
  id INT NOT NULL,  
  fname VARCHAR(30),  
  lname VARCHAR(30),  
  hired DATE NOT NULL DEFAULT '1970-01-01',  
  separated DATE NOT NULL DEFAULT '9999-12-31',  
  job_code INT NOT NULL,  
  store_id INT NOT NULL  
)  
PARTITION BY RANGE (store_id) (  
  PARTITION p0 VALUES LESS THAN (6),  
  PARTITION p1 VALUES LESS THAN (11),  
  PARTITION p2 VALUES LESS THAN (16),  
  PARTITION p3 VALUES LESS THAN  
  MAXVALUE  
);
```

- **List Partitioning, atau pemartisian daftar.**

Merupakan bentuk pemartisian yang sama dengan *Range Partitioning* hanya saja ekspresi jarak pemartisian yang diberikan dilakukan dengan pernyataan eksplisit. Contoh :

```
CREATE TABLE employees (  
  id INT NOT NULL,  
  fname VARCHAR(30),  
  lname VARCHAR(30),  
  hired DATE NOT NULL DEFAULT '1970-01-01',  
  separated DATE NOT NULL DEFAULT '9999-12-31',  
  job_code INT,  
  store_id INT  
)  
PARTITION BY LIST(store_id) (  
  PARTITION pNorth VALUES IN (3,5,6,9,17),  
  PARTITION pEast VALUES IN  
(1,2,10,11,19,20),  
  PARTITION pWest VALUES IN (4,12,13,14,18),  
  PARTITION pCentral VALUES IN (7,8,15,16)  
);
```

- **Hash partitioning, atau pemartisian dengan hash.**

Merupakan bentuk pemartisian ini sebenarnya sama dengan kedua bentuk pemartisian sebelumnya hanya saja pengguna tidak perlu menspesifikasikan ekspresi pemartisian dan cukup menunjuk atribut sebagai dasar dimana partisi-partisi akan dibentuk. Contoh :

```
PARTITION BY HASH(store_id)  
PARTITIONS 4 ;
```

- **Key partitioning, atau pemartisian dengan kunci.**

Merupakan bentuk pemartisian yang otomatis sepenuhnya. Seperti halnya pemartisian dengan hash pengguna tidak perlu menentukan ekspresi pemartisian. Contoh :

```
CREATE TABLE k1 (  
  id INT NOT NULL PRIMARY KEY,  
  name VARCHAR(20)  
)  
PARTITION BY KEY()  
PARTITIONS 2;
```

HASIL DAN PEMBAHASAN

Uji coba pemartisian dilakukan dengan 9 macam kueri yang memiliki karakteristik beda satu sama lainnya. Karakteristik kueri-kueri ialah pada tabel 1.

Tabel 1. Tabel Karakter Kueri Tes

Kueri	Join Tabel	Select	Where Clause	Order By
1	2 tabel	Selektif	tidak ada	NIP, Periode
2	tidak ada	Selektif	tidak ada	NIP, Periode
3	2 tabel	Selektif	periode = 2011	NIP, Periode
4	2 tabel	Selektif	less than 2006	NIP, Periode
5	2 tabel	Selektif	less than 2003	NIP, Periode
6	2 tabel	Selektif	jum.anak < 2	NIP, Periode
7	2 tabel	Selektif	jum.anak >= 2	NIP, Periode
8	3 tabel	Selektif	tidak ada	NIP, Periode
9	3 tabel	Selektif	tahun > 2006	NIP, Periode

hasil pengujian kecepatan kueri berdasarkan jumlah partisi dilakukan dengan hasil jumlah kueri sebagai berikut;

Tabel 2. Jumlah Data Hasil Kueri

Ku	Jum Data	Ku	Jum Data	Ku	Jum Data
1	477504	4	238752	7	320832
2	477504	5	119376	8	477504
3	39792	6	155672	9	253436

Uji coba dilakukan pada 12, 6, 3 dan tanpa partisi dengan hasil uji sebagai berikut;

Tabel 3. Uji Kueri 12 Partisi

Kueri	Waktu Kueri partisi 12 (Periode tahun)		
	Uji 1	Uji 2	Uji 3
1	16,578	16,574	16,141
2	13,751	13,813	13,547
3	0,594	0,513	0,594
4	7,895	7,781	8,812
5	2,016	2,156	2,344
6	14,501	13,906	14,675
7	14,872	15,516	14,973
8	15,078	15,281	14,996
9	4,687	4,611	4,641

Tabel 4. Uji Kueri 6 Partisi

Kueri	Waktu Kueri partisi 6 (Periode tahun)		
	Uji 1	Uji 2	Uji 3
1	17,031	17,563	16,819
2	14,578	14,312	14,473
3	0,547	0,547	0,531
4	7,751	8,781	8,782
5	2,157	2,251	2,21
6	14,334	14,453	14,657
7	14,719	15,344	15,172
8	16,607	16,047	16,703
9	6,594	6,609	6,187

Tabel 5. Uji Kueri 3 Partisi

Kueri	Waktu Kueri partisi 3 (Periode tahun)		
	Uji 1	Uji 2	Uji 3
1	17,091	16,781	17
2	14,5	14,344	14,156
3	0,51	0,547	0,55
4	8,789	8,719	8,781
5	2,359	2,641	2,625
6	2,5	2,406	2,24
7	15,078	15,28	14,891
8	17,547	17,487	17,642
9	8,021	6,987	6,907

Tabel 6. Uji Kueri tanpa Partisi

Kueri	Waktu Kueri tanpa partisi		
	Uji 1	Uji 2	Uji 3
1	51	60,422	57,156
2	36,219	33,672	33,873
3	0,53	0,51	1,43
4	24,469	24,579	24,89
5	4,469	4,89	4,906
6	53,672	53,657	49,156
7	55,61	54,984	56,61
8	39,78	37,985	39,812
9	9,46	10,12	10,547

Hasil Koefisien r ialah ;

Tabel 7. r Korelasi Hasil Uji

Ku	r korelasi	Ku	r korelasi	Ku	r korelasi
1	-0,6878	4	-0,703457	7	-0,68235
2	-0,70205	5	-0,75023	8	-0,74278
3	-0,29568	6	-0,514625	9	-0,93941

SIMPULAN

Untuk menyelesaikan pekerjaan penggajian dan pengupahan dengan kuantitas yang besar diperlukan sistem terkomputerisasi . Dari hasil uji statistik terhadap 9 macam kueri didapati bahwa rata-rata terdapat hubungan antara pemartisian basis data dengan kecepatan kueri dan kecepatan kueri cenderung berbanding terbalik dengan jumlah partisi dalam tabel(ditunjukkan dengan minus r).

DAFTAR PUSTAKA

- McCabe, JamesD., 2003, *Network analysis, architecture and design* , Morgan Kaufmann, San Francisco
- MySQL, Documentation Library, 2011, *MySQL 5.1 Reference Manual*, 08 Februari 2011. <<http://dev.mysql.com/doc/refman/5.1/en/partitioning-maintenance.html>>

STIKOM SURABAYA