OPTIMASI QUERY PADA LAPORAN TRANSAKSI PENJUALAN MENGGUNAKAN MATERIALIZED VIEW (Studi Kasus : Moonly café)

Febrianta Surya Nugraha[1]
Febryan Hari Purwanto[2]
Fachruddin Edi Nugroho Saputro[3]
[1,3]Mahasiswa Magister Teknik Informatika Universitas AMIKOM Yogyakarta
[2]Akademi Farmasi Al-Fatah Bengkulu
E-mail: [1]ubingg@gmail.com,
[2]fharipurwanto@gmail.com,
[3]fensdin@gmail.com,

Diterima: 2 Januari 2018/ Disetujui : 19 Januari 2018

ABSTRACT

The development of café business in Yogyakarta in 2017 is growing very rapidly. Moonly café currently uses sales information system using android platform and use data storage through cloud computing. The growing business café, transactions are done every day more and more. Sales information system runs slower when accessing larger data. These problems illustrate the lack of efficient time in data processing and the necessity of using query optimization in accessing data to speed up the process. The purpose of this study is to determine the speed performance in displaying monthly sales report by using query where clause, view table and materialized view table. The database is using MySql Server accessed via the internet, then recorded for the monthly sales report. In addition to comparing the performance of speed in displaying monthly sales reports, the authors also compare the speed of queries in displaying the report the amount of sales of each item each month on the table results of sales reports generated either from the query, view tables or materialized view tables. Use of materialized view to display monthly sales transaction reports faster than using query where clause or view. The use of materialized view to display reports the number of sales of each item each month faster than using the query where clause and view.

Keyword *: Database, Query optimization, Materialized view*

ABSTRAK

Perkembangan bisnis café di Yogyakarta tahun 2017 berkembang sangat pesat. Moonly café saat ini menggunakan sistem informasi penjualan menggunakan platform android serta menggunakan data penyimpanan melalui cloud computing. Semakin berkembangnya bisnis café, transaksi yang dilakukan setiap harinya semakin besar. Sistem informasi penjualan berjalan lebih lambat ketika mengakses data yang semakin besar. Dari permasalahan tersebut menggambarkan kurangnya efisien waktu dalam pemrosesan data dan perlunya penggunaan optimasi query dalam pengaksesan data untuk mempercepat proses tersebut. Tujuan dari penelitian ini adalah untuk mengetahui performa kecepatan dalam menampilkan laporan penjualan bulanan dengan menggunakan query where clause, tabel view dan tabel materialized view. Database menggunakan MySql server yang diakses melalui internet, kemudian dicatat waktunya untuk proses menampilkan laporan penjualan bulanan. Selain membandingkan performa kecepatan dalam menampilkan laporan penjualan bulanan, penulis juga membandingkan kecepatan query dalam menampilkan laporan jumlah penjualan masing-masing item tiap bulan pada tabel hasil laporan penjualan yang dihasilkan baik dari query, tabel view maupun tabel materialized view. Penggunaan materialized view untuk menampilkan laporan transaksi penjualan bulanan lebih cepat dari pada menggunakan query where clause maupun view.

Penggunaan materialized view untuk menampilkan laporan jumlah penjualan masing-masing item tiap bulan lebih cepat dari pada menggunakan query where clause maupun view.

***Kata kunci**: Database, Optimasi query, Materialized view*

1. PENDAHULUAN

Perkembangan bisnis café di Yogyakarta tahun 2017 berkembang sangat pesat. Penduduk yang heterogen dan kebiasaan masyarakat Yogyakarta yang sebagian besar dari kalangan mahasiswa mempunyai kebiasaan nongkrong serta diskusi di café. Semakin maraknya café-café di Yogyakarta membuat pengelola café harus berinovasi dengan menu-menu yang disajikan. Semakin banyaknya menu yang disajikan dengan kombinasi berbagai bahan baku yang semakin banyak, membuat proses-proses yang dilakukan di cafe secara manual berjalan dengan lambat. Oleh sebab itu, pengelola café mulai berfikir untuk menggunakan sistem informasi penjualan.

Dengan meningkatnya mobilitas masyarakat, pemilik café ingin dapat melihat dan memantau transaksi secara on-line atau dengan tidak perlu berada di café dengan memanfaatkan teknologi komputasi awan (cloud computing). Komputasi awan atau cloud computing dapat diartikan sebagai suatu teknologi yang memanfaatkan internet sebagai *resource* untuk komputasi yang dapat di-*request* oleh pengguna dan merupakan sebuah layanan dengan pusat server bersifat virtual atau berada dalam *cloud* (internet) itu sendiri [1].

Sistem informasi penjualan saat ini sudah berkembang dengan menggunakan data penyimpanan pada cloud computing. Semakin berkembangnya bisnis café, transaksi yang dilakukan setiap harinya semakin besar. Sistem informasi penjualan berjalan lebih lambat ketika mengakses data yang semakin besar. Dari permasalahan tersebut menggambarkan kurangnya efisien waktu dalam pemrosesan data dan perlunya penggunaan optimasi query dalam pengaksesan data untuk mempercepat proses tersebut.

View merupakan cara efisien untuk merepresentasikan data tanpa harus memeliharanya. View bukan merupakan tabel, dan bersifat virtual yang tidak membutuhkan storage tetap. View memiliki baris dan kolom seperti tabel, namun untuk dapat melakukan insert, update atau delete tergantung pada definisi view itu sendiri. Penggunaan view biasanya ditujukan untuk alasan keamanan data yang sensitif. Selain itu view juga dapat digunakan untuk mengubah tabel tanpa mempengaruhi aplikasi, memungkinkan untuk mengakses informasi lebih dari satu sumber, dan melakukan summary terhadap tabel seperti sum, min, max atau rata-rata [2]. Materialized view adalah tabel dinamis yang tidak hanya berisi perintah query SQL untuk menghasilkan baris, tetapi menyimpan baris yang sebenarnya. Materialized view dibuat saat pertama kali query dijalankan, dan baris ringkasan disimpan dalam tabel. Baris materialized view secara otomatis diperbarui saat tabel dasar diperbarui [3].

Moonly Café merupakan salah satu usaha kuliner di Yogyakarta yang beralamat di Jl. Menteri Supeno No 101, Umbulharjo Yogyakarta. Moonly Café berdiri sejak 1 April 2016 dan mempunyai 8 karyawan. Target pasar dari Moonly Café yaitu mahasiswa, siswa dan karyawan yang berada di Yogyakarta. Moonly cafe saat ini menggunakan sistem informasi penjualan menggunakan platform android serta menggunakan data penyimpanan melalui cloud computing. Laporan penjualan digunakan untuk pengontrolan penjualan setiap harinya oleh manajer operasional. Pengontrolan penjualan berkaitan langsung dengan stok dan bahan baku yang harus dibelanjakan setiap harinya. Selain itu laporan juga berfungsi untuk mengetahui *marketshare* penjualan, mengetahui grafik penjualan berdasarkan target pasar yg telah ditentukan, dan dapat berfungsi sebagai bahan evaluasi kinerja karyawan serta mengetahui selera pasar untuk inovasi kedepan yang disesuaikan dengan keinginan pelanggan. Selain manajer operasional, laporan penjualan digunakan pemilik cafe untuk evaluasi secara keseluruhan.

Penelitian terdahulu tentang *materializedview*antara lain adalah*an approach for query optimization by using schema object base view* [4], kompleksitas query sangat meningkatkan biaya eksekusi dari query yang memiliki efek kritis terhadap kinerja dan produktivitas sistem pendukung keputusan. Hal ini diperlukan untuk melakukan operasi penggabungan dan penggabungan berat yang sering dilakukan di database. Jika tidak dihitung sebelumnya, maka akan mengurangi kinerja query. Objek skema tidak hanya mendefinisikan hubungan, tapi juga memungkinkan untuk menghitung ulang join dan agregat yang besar yang menghasilkan kinerja query yang optimal. Objek skema

menyebabkan penurunan biaya pemrosesan query dan biaya pemeliharaan query dalam hal faktor waktu. Objek skema meningkatkan kinerja query dengan menghitung operasi penggabungan dan penggabungan yang berat pada database sebelum eksekusi dan menyimpan hasilnya di database. Keuntungan besar dari pandangan berbasis objek skema adalah pengambilan data agregat yang sangat cepat, karena dihitung dan disimpan, dengan mengorbankan insert / update / delete sehingga meningkatkan kinerja query daripada *view* dan tabel biasa. Hasil dari penelitian ini adalah kegunaan MV dalam menulis ulang query dan mengusulkan algoritma penulisan ulang yang terdiri dari tiga langkah utama. Pada langkah pertama, ia memilih MV yang akan digunakan untuk menulis ulang dan menentukan daerah query tersebut. Pendekatan sebelumnya berfokus pada optimalisasi query menggunakan agregasi dan bekerja pada satu blok query sementara di sini penulis menyajikan kegunaan skema objek sebagai tambahan dengan pekerjaan yang ada dan bekerja pada query multi blok. Pada langkah kedua, ia menghasilkan blok query untuk MV yang dipilih menggunakan wilayah query tersebut. Langkah terakhir mengintegrasikan blok query ke dalam query terakhir yang dapat ditulis Ini menggunakan kelas MV yang jauh lebih luas dan menghasilkan jenis penulisan ulang yang lebih umum daripada pendekatan sebelumnya.

Optimasi query database menggunakan teknik heuristik[5],heuristic *optimization*atau yang biasanya disebut dengan *rulebasedoptimization* adalah optimisasi query dengan menggunakan aturan-aturan heuristik dan dijalankan pada *logical* query *plan* (rencana query secara logika) yang terdiri dari urutan operasi-operasi relasional yang biasanya digambarkan sebagai query tree. Query *optimizer* mendapatkan sebuah inisial plan dari parser dan menggunakan aturan-aturan heuristik untuk mentransformasikan sebuah query ke dalam sebuah bentuk yang sama sehingga dapat diproses dengan lebih efisien. Hasil dari penelitian ini adalah optimisasi query berhubungan dengan teknik-teknik yang digunakan oleh DBMS (Database Manajemen Sistem) untuk memproses query agar diperoleh hasil query dengan waktu akses yang minimum. Terdapat 4 tahap yang harus dilakukan dalam optimisasi sebuah query, yaitu memasukkan query ke dalam representasi internal berdasarkan ekspresi aljabar yang sesuai; Mengkonversikannya ke dalam bentuk *canonical* dengan cara mula-mula dengan menggunakan *cartesianproduct* dari klausa FROM kemudian menggabungkan dan memilih kondisi-kondisi dari klausa WHERE dan melakukan proyeksi-proyeksi dari klausa SELECT; Memilih calon-calon prosedur *lowlevel*, yaitu mempertimbangkan *index* atau jalan akses lainnya, membagi nilai-nilai penyimpanan data dari *record* untuk memilih satu atau lebih calon-calon prosedur untuk mengimplementasikan tiap-tiap operasi *lowlevel* dalam query; Menghasilkan rencana-rencana query dan memilih yang termurah, yaitu membuat sekumpulan calon rencana-rencana query dan kemudian memilih yang termurah.

Design and implementation of algorithms for materialized view selection and maintenance in data warehousing environment [6],data *warehouse*, data mining dan pemrosesan analisis online merupakan beberapa tren terbaru dalam lingkungan komputasi dan aplikasi teknologi informasi untuk pemrosesan dan analisis data berskala besar. Teknologi data *warehouse* menjadi penting untuk perumusan strategi dan perumusan strategi bisnis yang efektif. Untuk keberhasilan setiap data warehouse, informasi konsolidasi yang akurat dan tepat waktu bersamaan dengan waktu respons query yang cepat dan efektif merupakan persyaratan dasar yang mendasar. *Materializationviews* secara praktis tidak mungkin karena adanya ruang penyimpanan dan batasan biaya pemeliharaan yang terwujud sehingga pemilihan materialized view dengan baik merupakan salah satu keputusan cerdas dalam merancang gudang data untuk mendapatkan efisiensi yang optimal. Dalam makalah ini, penulis menyajikan kerangka kerja untuk memilih *materializedview* terbaik sehingga dapat mencapai kombinasi yang efektif dari waktu respons query yang baik, biaya pemrosesan query yang rendah dan biaya perawatan tampilan rendah dalam batasan ruang penyimpanan tertentu. Parameter implementasi kerangka kerja mencakup biaya frekuensi query, biaya penyimpanan query dan biaya pemrosesan query. Kerangka kerja penelitian ini memilih biaya terbaik yang efektif padamaterlized view untuk mengoptimalkan waktu pemrosesan query sehingga menghasilkan sistem data *warehouse* yang efisien.

Tujuan dari penelitian ini adalah untuk mengetahui performa kecepatan dalam menampilkan laporan penjualan bulanan dengan menggunakan query where clause, tabel view dan tabel materialized view. Database diletakan pada server yang diakses melalui internet, kemudian dicatat waktunya untuk proses menampilkan laporan penjualan bulanan. Selain membandingkan performa kecepatan dalam menampilkan laporan penjualan bulanan, penulis juga membandingkan kecepatan

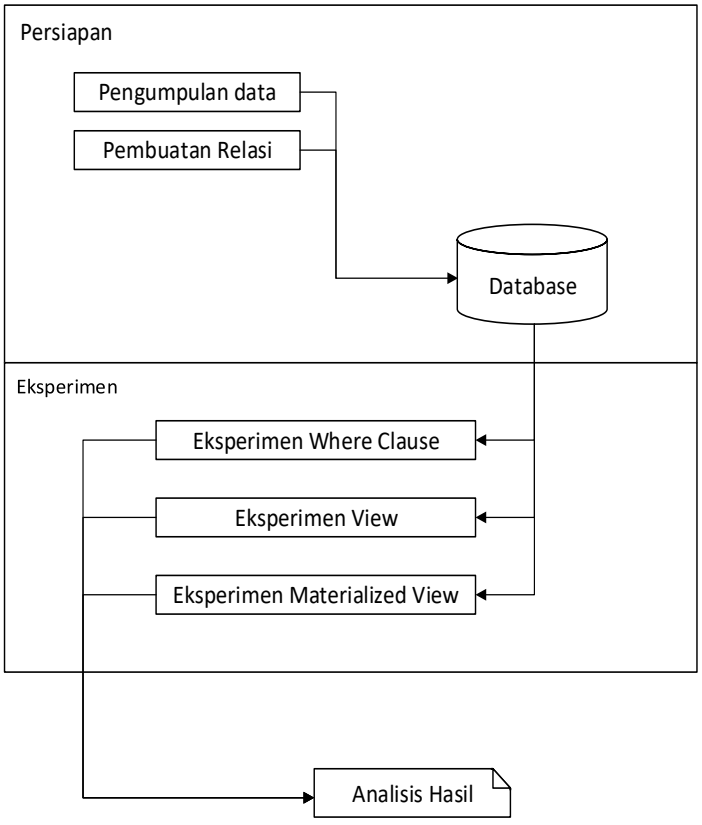
query dalam menampilkan laporan jumlah penjualan masing-masing item tiap bulan pada tabel hasil laporan penjualan yang dihasilkan baik dari query*whereclause*, tabel *view* maupun tabel *materializedview*.

2. METODE PENELITIAN

Penelitian ini terdiri dari 3 tahapan, yaitu tahap persiapan, eksperimen dan analisis hasil. Pada tahap persiapan, dimulai dengan mengumpulkan beberapa data transaksi penjualan pada Moonly Café kemudian dilakukan normalisasi untuk membuat relasi tabel. Setelah itu dilakukan pembuatan *database* di server dari data transaksi penjualan dan relasi tabel.

Pada tahap eksperimen, dilakukan eksperimen pembuatan dan menampilkan transaksi penjualan bulanan dengan menggunakan operasi *whereclause,view* dan *materializedview*kemudian dicatat lamanya waktu proses. Pada tahap eksperimen juga dilakukan eksperimen query untuk menampilkanlaporan jumlah penjualan masing-masing item tiap bulan dari hasil menanpilkan penjualan bulanan dengan menggunakan operasi *where clause*, *view* dan *materializedview*.

Pada tahap analisis hasil, dilakukan pembuatan rekapitulasi hasil dari eksperimen yang dilakukan, kemudian dibandingkan hasil dari eksperimen tersebut. Alur penelitian dapat dilihat pada gambar 1.



Gambar 1. *Alur Penelitian*

3. HASIL DAN PEMBAHASAN

3.1 Tahap Persiapan

Pada tahap persiapan, dimulai dengan mengumpulkan beberapa data transaksi penjualan pada Moonly Café kemudian dilakukan normalisasi untuk membuat struktur tabel. Dari struktur tabel dibuat relasi tabel yang akan digunakan untuk pembuatan database. Setelah database dibuat, data yang semula berbentuk excel perlu diubah menjadi csv supaya dapat dimasukan kedalam database. Tahap terakhir sebelum melakukan eksperimen adalah pengecekan konektifitas komputer dan server yang digunakan untuk eksperimen.

Nomor	Tanggal	Pelanggan	Item	Qty	Harga Satuan	Jumlah	Total
S/170409/1	2017-04-09, 17:13	-	Teh Panas	1	Rp. 4.000	Rp. 4.000	Rp. 4.000
			French Fries	1	Rp. 8.000	Rp. 8.000	
			Jamur Pedas Cryspi	1	Rp. 7.000	Rp. 7.000	
			Mie Telur Goreng	1	Rp. 9.000	Rp. 9.000	
			Milkshake Coklat	1	Rp. 11.000	Rp. 11.000	
S/170409/2	2017-04-09, 19:05	-	Milkshake Vanilla	1	Rp. 11.000	Rp. 11.000	Rp. 46.000
			Air Mineral	2	Rp. 4.000	Rp. 8.000	
			Ice Cream Vanilla	1	Rp. 8.000	Rp. 8.000	
			Jus Alpukat	1	Rp. 10.000	Rp. 10.000	
			Jus Mangga	2	Rp. 10.000	Rp. 20.000	
			Lemon Squash	2	Rp. 8.000	Rp. 16.000	
			Lemon Tea	1	Rp. 6.000	Rp. 6.000	
S/170409/3	2017-04-09, 19:22	-	Air Putih	2	Rp. 1.000	Rp. 2.000	Rp. 70.000
			Cappucino	1	Rp. 7.000	Rp. 7.000	
			Susu Jahe	1	Rp. 7.000	Rp. 7.000	
			Vanila Late	1	Rp. 6.000	Rp. 6.000	
S/170409/4	2017-04-09, 19:29	-	Vanila Late	1	Rp. 6.000	Rp. 6.000	Rp. 26.000
			Milkshake Vanilla	1	Rp. 11.000	Rp. 11.000	
S/170409/5	2017-04-09, 19:31	-	Pisang Goreng Krispy	1	Rp. 6.000	Rp. 6.000	Rp. 17.000
			Air Mineral	1	Rp. 4.000	Rp. 4.000	
			French Fries	1	Rp. 8.000	Rp. 8.000	
			Jeruk Panas	1	Rp. 6.000	Rp. 6.000	
			Jus Alpukat	1	Rp. 10.000	Rp. 10.000	
			Jus Mangga	1	Rp. 10.000	Rp. 10.000	
			Lemon Squash	2	Rp. 8.000	Rp. 16.000	
S/170409/6	2017-04-09, 19:44	-	Strawberry Float	1	Rp. 13.000	Rp. 13.000	Rp. 67.000
			Air Putih	1	Rp. 1.000	Rp. 1.000	

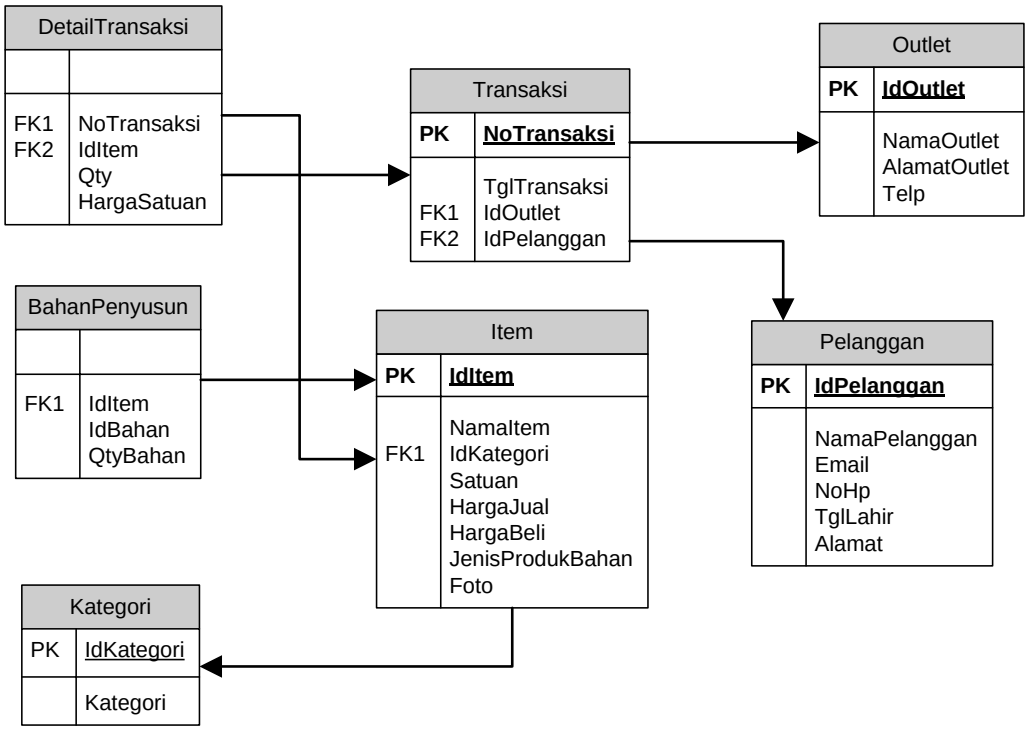
Gambar 2. *Contoh Data Laporan Penjualan Moonly Cafe*

Data transaksi penjualan pada Moonly cafe yang berbentuk excel dapat dilihat pada gambar 2. Dari data transaksi laporan penjualan, dilakukan normalisasi sehingga didapatkan struktur tabel database yang dapat dilihat pada tabel 1.

Tabel 1. Struktur Tabel database

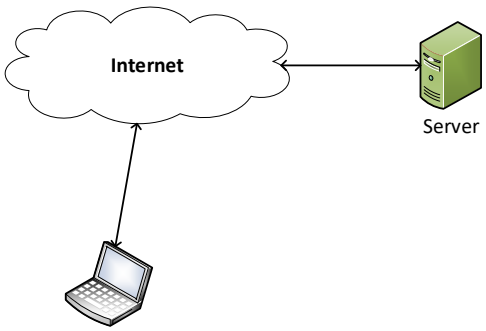
Tabel bahanpenyusun		
Nama	Jenis	Deskripsi
IdItem (FK)	varchar(5)	Kode barang
IdBahan	varchar(5)	Kode bahan penyusun
QtyBahan	int(10)	Jumlah bahan Penyusun
Tabel detailtransaksi		
Nama	Jenis	Deskripsi
NoTransaksi (FK)	varchar(20)	Nomor Nota
IdItem (FK)	varchar(5)	Kode barang
Qty	int(10)	Jumlah pembelian
HargaSatuan	int(20)	Harga satuan
Tabel item		
Nama	Jenis	Deskripsi
IdItem (PK)	varchar(5)	Kode barang
NamaItem	varchar(50)	Nama barang
IdKategori (FK)	varchar(5)	Kode kategori barang
Satuan	varchar(5)	Satuan
HargaJual HargaBeli	int(20) int(20)	Harga jual Harga beli
JenisProdukBahan	varchar(10)	Jenis produk atau bahan
Foto	varchar(255)	Gambar produk
Tabel kategori		
Nama	Jenis	Deskripsi
IdKategori (PK)	varchar(5)	Kode kategori barang
Kategori	varchar(30)	Nama kategori
Tabel outlet		
Nama	Jenis	Deskripsi
IdOutlet(PK)	varchar(5)	Kode outlet
NamaOutlet	varchar(40)	Nama outlet
AlamatOutlet	varchar(100)	Alamat outlet
Telp	varchar(15)	Nomor Telepon
Tabel pelanggan		
Nama	Jenis	Deskripsi
IdPelanggan (PK)	varchar(5)	Kode pelanggan
NamaPelanggan	varchar(30)	Nama pelanggan
Email	varchar(50)	Email pelanggan
NoHp	varchar(15)	Nomor handphone pelanggan
TglLahir	date	Tanggal lahir pelanggan
Alamat	varchar(100)	Alamat pelanggan
Tabeltransaksi		
Nama	Jenis	Deskripsi
NoTransaksi (PK)	varchar(20)	Nomor transaksi
TglTransaksi	date	Tanggal transaksi
IdOutlet (FK)	varchar(5)	Kode outlet
IdPelanggan (FK)	varchar(5)	Kode pelanggan

Dari beberapa tabel yang dihasilkan, untuk melihat hubungan antar tabel dibuatlah relasi tabel. Tanda PK merupakan kunci utama, sedangkan tanda FK adalah kunci tamu yang ditentukan oleh arah panah. Relasi tabel dapat dilihat pada gambar 3.



Gambar 3. *Relasi Tabel*

Berdasarkan relasi tabel dan data-data yang sudah diubah kedalam format csv, dibuatlah database yang terletak pada server cloud yang diakses melalui internet. DBMS (*database management system*) yang digunakan adalah MySQL yang diakses menggunakan web *browser* melalui phpmyadmin.Gambaran umum topologi dapat dilihat pada gambar 4.



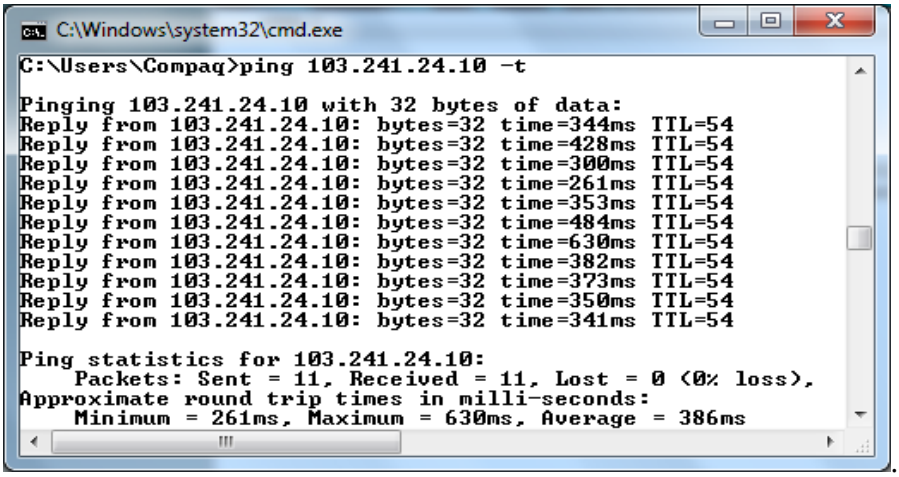
Gambar 4. *Gambaran Umum Topologi*

Database yang dibuat dari hasil normalisasi sebelumnya diletakan pada server yang di akses melalui internet yang beralamat pada https://nazira.kingofserver.net dengan konfigurasi sebagai berikut.

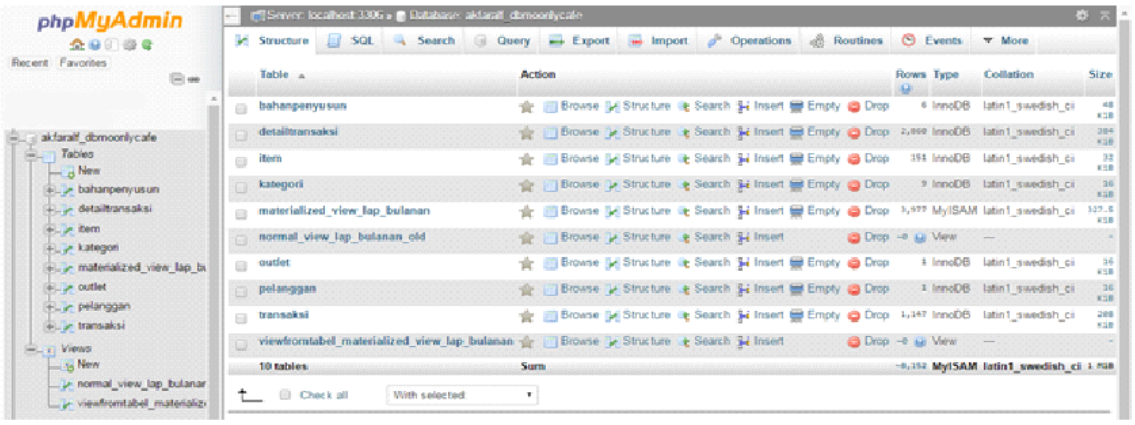
Database server
Server: Localhost via UNIX socket
Server type: MySQL
Server version: 5.6.35-cll-lve - MySQL Community Server (GPL)

Protocol version: 10
User: akfaralf_moonly@localhost
Database Name :akfaralf_dbmoonlycafe
Server charset: UTF-8 Unicode (utf8)
Web server
cpsrvd 11.64.0.22
Database client version: libmysql - 5.1.73
PHP extension: mysqliDocumentation curlDocumentation mbstringDocumentation
PHP version: 5.6.30
phpMyAdmin Version information: 4.6.6

Untuk membandingkan performa kecepatan penggunaan query where clause, tabel view dan tabel materialized view, penulis menggunakan tool phpmyadmin versi 4.6.6 yang terinstall pada server. Untuk mengakses phpmyadmin dan menjalankan perintah query digunakan browser Google Chrome Versi 58.0.3029.110 yang terinstal pada notebook dengan spesifikasi Intel DualCore T3400 2.16GHz, 2048MB DDR2, dan sistem operasi Windows 7 Ultimate 32 Bit. Koneksi yang digunakan menggunakan koneksi internet dengan jaringan HSDPA+ dan nilai ping ke server saat penelitian adalah nilai minimum 261ms, maximum 630ms,, dan rata-rata 386ms yang dapat dilihat pada gambar 5 dan tampilan phpmyadmin pada gambar 6.



Gambar 5. *Hasil Ping ke Server*



Gambar 6. *Tampilan PhpMyAdmin Server*

3.2 Eksperimen

Setelah dilakukan import struktur dan data ke database, dilakukan pengujian untuk mengetahui performa kecepatan dalam menampilkan data laporan penjualan bulanan dan

menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentudengan menggunakan query*where clause*, *view*, dan *materialized view*. Masing-masing pengujian dilakukan sebanyak 5 kali pengujian yang terdiri dari pengujian dengan 100, 500, 750, 1000, dan 1500 baris data.

3.2.1 Eksperimen Menggunakan Where Clause

Query yang digunakan dalam eksperimen menampilkan laporan transaksi penjualan bulanan menggunakan query *where clause* dapat dilihat pada query 1 dan menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu pada query 2.

Query 1. Menampilkan Laporan Bulanan Dengan Query *Where Clasue*

```
SELECT
o.NamaOutlet,p.NamaPelanggan,COALESCE(t.TglTransaksi,'GRAND
TOTAL') AS TglTransaksi,COALESCE(t.NoTransaksi,'TOTAL HARIAN') AS
NoTransaksi,COALESCE(i.NamaItem,'TOTAL PER TRX') AS NamaItem
,dt.Qty,dt.HargaSatuan, SUM(dt.Qty*dt.HargaSatuan) AS JumlahdanTotal
FROM
outlet o,transaksi t,pelanggan p,item i,detailtransaksi dt
WHERE
o.IdOutlet=t.IdOutlet AND
p.IdPelanggan=t.IdPelanggan AND
t.NoTransaksi=dt.NoTransaksi AND
i.IdItem=dt.IdItem AND
YEAR(t.TglTransaksi)='2017' AND MONTH(t.TglTransaksi)='05'
GROUP BY t.TglTransaksi,t.NoTransaksi,i.NamaItem WITH ROLLUP;
```

Query 2. Menampilkan Laporan Jumlah Penjualan Masing-Masing Item Pada Bulan Tertentu Dengan Query *WhereClause*

```
SELECT
o.NamaOutlet,i.NamaItem,COUNT(dt.iditem) AS JumlahPenjualan
FROM
outlet o,transaksi t,pelanggan p,item i,detailtransaksi dt
WHERE
o.IdOutlet=t.IdOutlet AND
p.IdPelanggan=t.IdPelanggan AND
t.NoTransaksi=dt.NoTransaksi AND
i.IdItem=dt.IdItem AND
YEAR(t.TglTransaksi)='2017' AND MONTH(t.TglTransaksi)='05'
GROUP BY i.NamaItem
ORDER By JumlahPenjualan DESC;
```

Hasil pengujian query 1 menampilkan laporan bulanan dan query 2 untuk menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu dengan query *whereclause* dapat dilihat pada tabel 2.

Tabel 2. Hasil Pengujian Menggunakan *Where Clause*

Parameter	Pengujian 100 Baris	Pengujian 500 Baris	Pengujian 750 Baris	Pengujian 1000 Baris	Pengujian 1500 Baris
Query 1	0.007	0.0072	0.0072	0.008	0.0076
Query 2	0.0039	0.0045	0.0037	0.0045	0.004

3.2.2 Eksperimen Menggunakan View

Sebelum dilakukan pengujian menampilkan laporan transaksi penjualan bulanan menggunakan *view*, dibuat tabel*view* terlebih dahulu dengan query sebagai berikut.

```
CREATE VIEW normal_view_lap_bulanan AS
SELECT
o.NamaOutlet,p.NamaPelanggan,t.TglTransaksi,t.NoTransaksi,i.NamaItem,dt.Qty
,dt.HargaSatuan, SUM(dt.Qty*dt.HargaSatuan) AS JumlahdanTotal
FROM
outlet o,transaksi t,pelanggan p,item i,detailtransaksi dt
WHERE
o.IdOutlet=t.IdOutlet AND
p.IdPelanggan=t.IdPelanggan AND
t.NoTransaksi=dt.NoTransaksi AND
i.IdItem=dt.IdItem
GROUP BY t.TglTransaksi,t.NoTransaksi,i.NamaItem WITH ROLLUP;
```

Query yang digunakan dalam eksperimen menampilkan laporan transaksi penjualan bulanan menggunakan *view* dapat dilihat pada query 3 dan menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu pada query 4.

Query 3. Menampilkan Laporan Bulanan Menggunakan*View*

```
SELECT * FROM normal_view_lap_bulanan
WHERE YEAR(TglTransaksi)='2017' AND MONTH(TglTransaksi)='05';
```

Query 4. Menampilkan Laporan Jumlah Penjualan Masing-Masing Item Pada Bulan Tertentu Menggunakan *View*

```
SELECT
NamaOutlet,NamaItem,COUNT(NamaItem) AS JumlahPenjualan
FROM
normal_view_lap_bulanan_old
WHERE NamaItem IS NOT NULL AND
YEAR(TglTransaksi)='2017' AND MONTH(TglTransaksi)='05'
GROUP BY NamaItem
ORDER By JumlahPenjualan DESC;
```

Hasil pengujian query 3 menampilkan laporan bulanan dan query 4 untuk menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu dengan menggunakan *view* dapat dilihat pada tabel 3.

Tabel 3. Hasil Pengujian Menggunakan *View*

Parameter	Pengujian 100 Baris	Pengujian 500 Baris	Pengujian 750 Baris	Pengujian 1000 Baris	Pengujian 1500 Baris
Query 3	0.0105	0.0104	0.0111	0.0118	0.0124
Query 4	0.0209	0.0174	0.017	0.0129	0.0113

3.2.3 Eksperimen Menggunakan Materialized View

Sebelum dilakukan pengujian menampilkan laporan transaksi penjualan bulanan menggunakan *materializedview*, dibuat tabel *materialized view* terlebih dahulu dengan query sebagai berikut.

```
CREATE TABLE materialized_view_lap_bulanan AS SELECT * FROM
normal_view_lap_bulanan;
RENAME TABLE normal_view_lap_bulanan TO normal_view_lap_bulanan_old;
CREATE VIEW viewfromtabel_materialized_view_lap_bulanan AS SELECT * FROM
materialized_view_lap_bulanan;
```

Query yang digunakan dalam eksperimen menampilkan laporan transaksi penjualan bulanan menggunakan *materializedview* dapat dilihat pada query 5 dan menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu pada query 6.

Query 5. Menampilkan Laporan Bulanan Menggunakan*MaterializedView*

```
SELECT * FROM viewfromtabel_materialized_view_lap_bulanan
WHERE YEAR(TglTransaksi)='2017' AND MONTH(TglTransaksi)='05';
```

Query 6. Menampilkan Laporan Jumlah Penjualan Masing-Masing Item Pada Bulan Tertentu Menggunakan *MaterializedView*

```
SELECT
NamaOutlet,NamaItem,COUNT(NamaItem) AS JumlahPenjualan
FROM
viewfromtabel_materialized_view_lap_bulanan
WHERE NamaItem IS NOT NULL AND
YEAR(TglTransaksi)='2017' AND MONTH(TglTransaksi)='05'
GROUP BY NamaItem
ORDER By JumlahPenjualan DESC;
```

Hasil pengujian query 5 menampilkan laporan bulanan dan query 6 untuk menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu dengan menggunakan *materializedview* dapat dilihat pada tabel 4.

Tabel 4. Hasil Pengujian Menggunakan *MaterializedView*

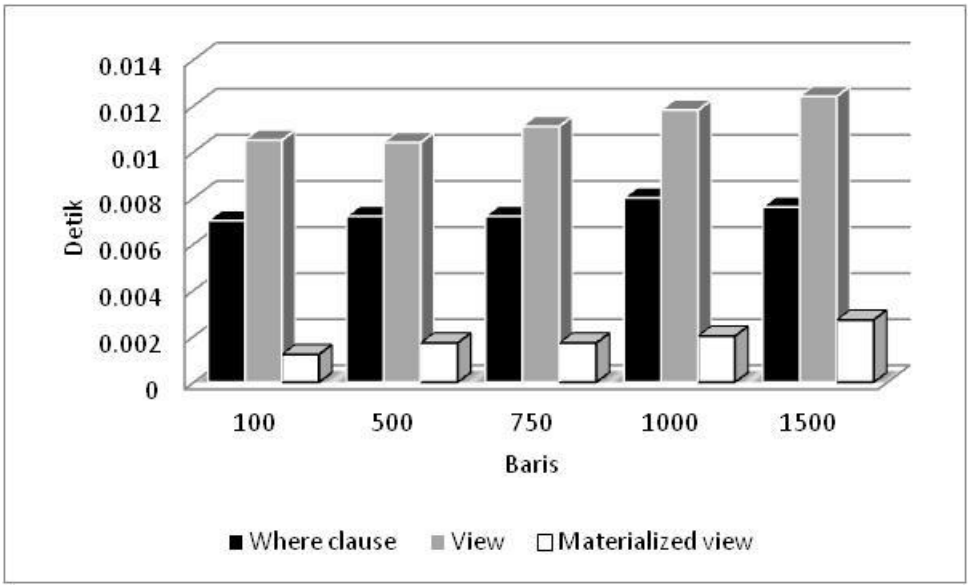
Parameter	Pengujian 100 Baris	Pengujian 500 Baris	Pengujian 750 Baris	Pengujian 1000 Baris	Pengujian 1500 Baris
Query 5	0.0012	0.0017	0.0017	0.002	0.0027
Query 6	0.0011	0.0022	0.0024	0.0024	0.0029

3.3 Analisis Hasil

Berdasarkan eksperimen yang talah dilakukan, untuk mengetahui performa kecepatan pemrosesan query, perlu dilakukan perbandingan hasil eksperimen baik menggunakan query *where clause*, *view*, dan *materializedview*. Perbandingan performa kecepatan pemrosesan query untuk menampilkan laporan transaksi penjualan bulanan dapat dilhat pada tabel 5 dan gambar 7.

Tabel 5. Hasil Perbandingan Menampilkan Laporan Transaksi Penjualan Bulanan

Parameter	Pengujian 100 Baris	Pengujian 500 Baris	Pengujian 750 Baris	Pengujian 1000 Baris	Pengujian 1500 Baris
Where clause(query 1)	0.007	0.0072	0.0072	0.008	0.0076
View(query 3)	0.0105	0.0104	0.0111	0.0118	0.0124
Materializedview(query 5)	0.0012	0.0017	0.0017	0.002	0.0027

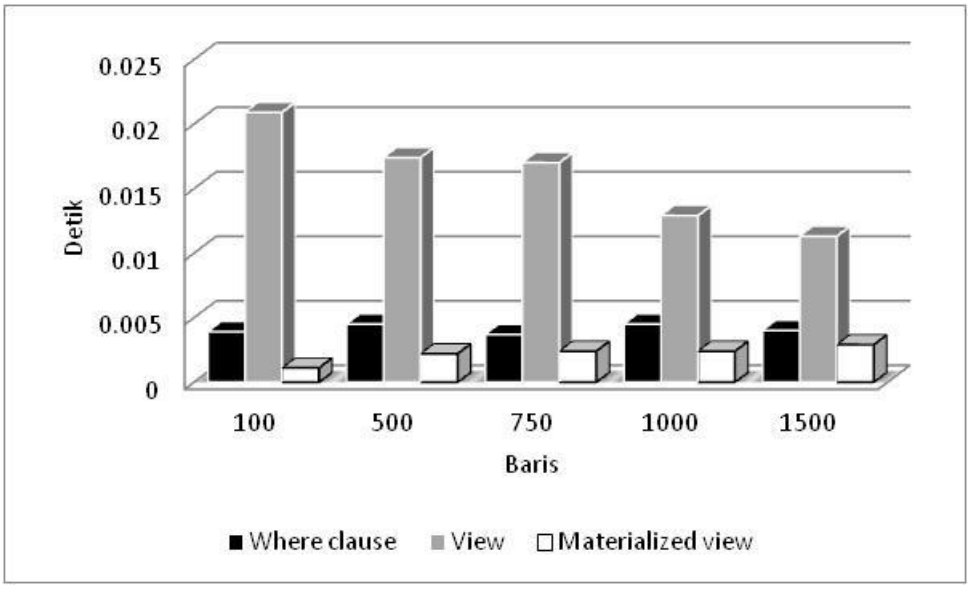


Gambar 7. *Grafik Perbandingan Menampilkan Laporan Transaksi Penjualan Bulanan*

Perbandingan performa kecepatan pemrosesan query untuk menampilkan laporan jumlah penjualan masing-masing item pada bulan tertentu dapat dilhat pada tabel 6 dan gambar 8.

Tabel 6. Hasil Perbandingan Menampilkan Laporan Jumlah Penjualan Masing-Masing Item

Parameter	Pengujian 100 Baris	Pengujian 500 Baris	Pengujian 750 Baris	Pengujian 1000 Baris	Pengujian 1500 Baris
Where clause(query 2)	0.0039	0.0045	0.0037	0.0045	0.004
View(query 4)	0.0209	0.0174	0.017	0.0129	0.0113
Materializedview(query 6)	0.0011	0.0022	0.0024	0.0024	0.0029



Gambar 8. *Grafik Perbandingan Menampilkan Laporan Jumlah Penjualan Masing-Masing Item*

Berdasarkan hasil perbandingan dalam menampilkan laporan transaksi penjualan bulanan dan laporan jumlah penjualan masing-masing item, penggunaan *materialized view* jauh lebih cepat dari pada menggunakan *where clause* maupun *view*. Waktu pemrosesan query pada *view* lebih besar dari pada menggunakan *whereclause*.

Setelah dilakukan beberapa eksperimen untuk membandingkan waktu pemrosesan query antara *where clause*, *view* dan *materialized view*, maka disarankan dilakukan optimasi query untuk meningkatkan kecepatan dalam menampilkan laporan penjualan bulanan, yang semula menggunakan *where clause* diganti menggunakan *materialized view*.

4. KESIMPULAN

Berdasarkan hasil pengujian diatas, dapat disimpulkan antara lain sebagai berikut.

1. Waktu pemrosesan query dengan menggunakan*where clause* untuk menampilkan laporan transaksi penjualan bulanandengan jumlah data 100, 500, 750, 1000, dan 1500 data membutuhkan waktu 0.007, 0.0072, 0.0072, 0.008 dan 0.0076 detik. Waktu pemrosesan query dengan menggunakan *where clause* untuk menampilkan laporan jumlah penjualan masing-masing item dengan jumlah data 100, 500, 750, 1000, dan 1500 data membutuhkan waktu 0.0039, 0.0045, 0.0037, 0.0045 dan 0.004 detik.
2. Waktu pemrosesan query dengan menggunakan *view* untuk menampilkan laporan transaksi penjualan bulanan dengan jumlah data 100, 500, 750, 1000, dan 1500 data membutuhkan waktu 0.0105, 0.0104, 0.0111, 0.0118 dan 0.0124 detik. Waktu pemrosesan query dengan menggunakan *view* untuk menampilkan laporan jumlah penjualan masing-masing item dengan jumlah data 100, 500, 750, 1000, dan 1500 data membutuhkan waktu 0.0209, 0.0174, 0.017, 0.0129 dan 0.0113 detik.

3. Waktu pemrosesan query dengan menggunakan *materializedview* untuk menampilkan laporan transaksi penjualan bulanan dengan jumlah data 100, 500, 750, 1000, dan 1500 data membutuhkan waktu 0.0012, 0.0017, 0.0017, 0.002 dan 0.0027 detik. Waktu pemrosesan query dengan menggunakan *materializedview* untuk menampilkan laporan jumlah penjualan masing-masing item dengan jumlah data 100, 500, 750, 1000, dan 1500 data membutuhkan waktu 0.0011, 0.0022, 0.0024, 0.0024 dan 0.0029 detik.
4. Penggunaan *materializedview* untuk menampilkan laporan transaksi penjualan bulanan dan laporan jumlah penjualan masing-masing item lebih cepat dari pada menggunakan query *where clause* maupun *view*, sehingga *Materialized view* dapat disarankan untuk digunakan sebagai salah satu metode optimasi pada proses penarikan data laporan transaksi penjualan bulanan di Moonly Café.

5. SARAN

Saran dari penelitian ini adalah membandingkan performa query *whereclause*, *view* dan *materializedview* dengan menggunakan data yang lebih besar dari 1500 data dan membandingkan dengan menggunakan parameter lain selain parameter waktu pemrosesan query.

DAFTAR PUSTAKA

[1] Jamil, M., Rosihan, dan Fuad, A., 2016, *Cloud Computing : Teori dan Aplikasi*, Deepublish, Yogyakarta.
[2] Sutiani, D., 2010, *IBM Database - DB2 for Beginners* , Elex Media Komputindo, Jakarta.
[3] Coronel, C., dan Morris, S., 2016, *Database Systems: Design, Implementation, and Management* , Cengage Learning, Boston, United States of America.
[4] Patel, D., dan Patel, P., 2015, An Approach for Query Optimization by using Schema Object Base View, *International Journal of Computer Applications*, Vol. 119, No. 16, 21-24, :http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.3161&rep=rep1&type=pdf.
[5] Sudaryanto, S. N., 2007, Optimasi Query Database Menggunakan Teknik Heuristic, *Techno.com*, Vol. 7, No. 2, 53-66, :http://dinus.ac.id/wbsc/assets/dokumen/majalah/Optimasi_Query_Database_Menggunakan_Teknik_Heuristic1.pdf.
[6] Jogekar. R. N., dan Mohod, A., 2013, Design and Implementation of Algorithms for Materialized View Selection and Maintenance in Data Warehousing Environment, *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, No. 9, 464-470, :https://pdfs.semanticscholar.org/6619/08d9000559b1aff343b66e047e8713daa8a3.pdf